64'er

SONDERHEFT
FLOPPY-LAUFWERKE

SONDERHEFT 25 ÖS 100,-/sfr. 14,- Lit. 12000/hfl. 18,-/dkr. 72,- DM 14,-

Markt&Technik

64'er

Ausführliche Kurse

★ Für Einsteiger und Profis: So wird die Floppy 1541 programmiert

★ 1571 intern: Geheimnisse enthüllt

★ Schnell und flexibel: Floppy-Zugriff unter Geos

Top-Listings

★ Floppy 1541: Superschnelles Kopierprogramm

★ Spitzen-Monitor für alle 1571-Formate

★ Mehr Speicherplatz auf Diskette mit neuem Directory

Hardware unter der Lupe

★ Aufgedeckt: Was ist anders bei den neuen Floppies?

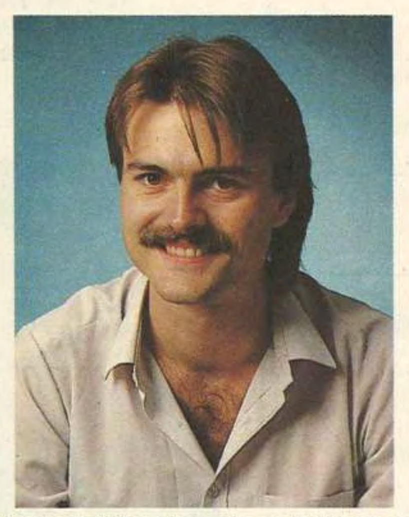
★ Im Test: Die Floppy 1581

★ Übersicht: Speeder für alle Laufwerke

Alle Programme auch auf Diskette erhältlich


GEOS

Höher, schneller, weiter



Was dem Sport recht ist, ist der Computerbranche billig. Überschlagen sich doch die Meldungen über immer neue Rekorde in bezug auf Geschwindigkeit, Leistungsfähigkeit und Speicherkapazitäten bei neuen Computern oder Peripheriegeräten fast täglich.

Von dieser Entwicklung bleiben auch die Heimcomputer nicht »verschont«. Bieten doch gerade Heimcomputer wie der C64, der C128 und ihre Peripheriegeräte ein ideales Feld für Tüfteleien und immer neue Verbesserungen. Keine Ausnahme machen hier die Floppy-Laufwerke. Sowohl auf dem Hard- als auch auf dem Software-Sektor tut sich hier einiges:

Die Zeiten, in denen die Überlegung schwerfiel, welche Art von Massenspeicher – Floppy oder Datasette – man sich für seinen C64 oder C128 zulegt, gehen ihrem Ende entgegen. Dies hat mehrere Gründe: Zum einen ist der Preis der Commodore-Laufwerke mittlerweile auf ein sehr attraktives Niveau gesunken, zum anderen erfordert die Verwendung leistungsfähiger Software einfach eine Floppystation. Geos, um nur ein Beispiel zu nennen, ist ein diskettenorientiertes Betriebssystem, das jegliche Zusammenarbeit mit der Datasette verweigert.

Mittlerweile steht zudem eine ganze Palette von Floppy-Laufwerken für den C64/ C128 zur Auswahl: Hier wäre zuerst die »gute alte« 1541 zu nennen, die inzwischen ein »Facelifting« zur 1541c hinter sich hat. Dies brachte leider keinen Geschwindigkeitszuwachs, dafür aber geringere Kompatibilität zur Vorgängerin und damit zu mancher Software.

Stichwort »kompatibel«: Das Floppy-Laufwerk im neuen C128D mit dem Blechgehäuse kann auch nicht gerade mit dieser Eigenschaft prahlen. Lesen Sie woran dies liegt, und warum deshalb viele Programme nicht laufen.

Vor kurzem kam die Floppy 1581 auf den Markt. Mit ihr beginnt sich ein neuer Standard zu etablieren: Die 3½- Zoll-Laufwerke sind im Kommen. Wir haben in diesem Sonderheft die Floppy 1581 für Sie getestet und ihren Konkurrenten gegenübergestellt.

Zudem hat sich um die Commodore-Floppies herum eine ganze Branche entwickelt, die es sich zur Aufgabe gemacht hat, diese Geräte auf immer höhere Niveaus der Leistung und Geschwindigkeit anzuheben.

Da die Floppy 1541, ebenso wie ihre Nachfolgerin, die 1541c, wahrlich nicht zur schnellen Truppe gehören, wundert es nicht, daß eine rege Nachfrage für Floppy-Beschleuniger und andere Hardware-Zusätze besteht.

Nicht viel anders sieht es aus mit der Floppy 1571, die am C128 betrieben, eine recht ordentliche Geschwindigkeit an den Tag legt. Doch selbst da schafften es findige Köpfe, noch mehr herauszuholen.

Was hat es nun mit den neusten Entwicklungen auf sich, was leisten diese, was kosten sie und wie kompatibel sind diese zur bestehenden Software?

Disketten-Monitore sind der unentbehrliche Helfer eines jeden Computer-Fans. Doch leider mußten diese bislang meist vor zwei Dingen passen:

Entweder, sie konnten die »illegalen Tracks« einer Diskette, nämlich die Spuren 36 bis 40, nicht lesen, oder aber sie versagten bei doppelseitig formatierten Disketten der Floppy 1571. Dem wird jetzt mit »Top-Flop« einem Disk-Monitor für den C128 und »Disc-Scanner 40« für den C64 abgeholfen.

Was man aus der Floppy 1541 (fast) ohne Hardware-Zusatz herausholen kann, zeigt das Super-Kopierprogramm »Master Copy Parallel«. Mit diesem Programm kann eine komplette Diskette in nur 18 Sekunden kopiert werden, wenn man die Diskettenwechsel nicht mitrechnet.

Mit »XDOS« erhalten Sie ein tolles Programm, das eine komfortable Benutzeroberfläche zur Bedienung aller Floppy-Funktionen darstellt. Durch Steuerung mit dem Joystick gehören damit endlose Befehlssequenzen an das Laufwerk der Vergangenheit an.

In ausführlichen Kursen erfahren Sie in diesem Sonderheft zudem alles über Funktionsweise und Aufbau und Programmierung der Floppy 1541, die ja eigentlich einen kompletten Computer darstellt.

Ins »Eingemachte« der Floppy 1571 geht dann ein Beitrag, der sich vor allem mit den schnellen Burst-Routinen dieser Diskettenstation befaßt. Dort lernen Sie, diese für eigene Programme zu nutzen. So »ganz nebenbei« beschert Ihnen dieser Artikel das Programm »Burstmon«, mit dem es sogar möglich ist, Disketten, die im MFM-Format (MS-DOS- oder CP/M-Computer verwenden dieses Diskettenformat) beschrieben sind, zu lesen.

Eine runde Sache also, dieses Sonderheft, aus dem jeder, sei es Einsteiger, Fortgeschrittener oder Profi, wertvolle Tips und Informationen ziehen kann.

Klaus Schrödl
(Redakteur)

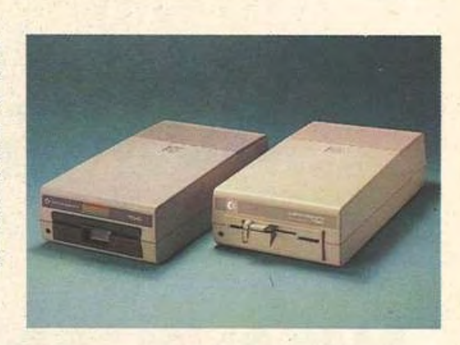
INHALT

Hardware

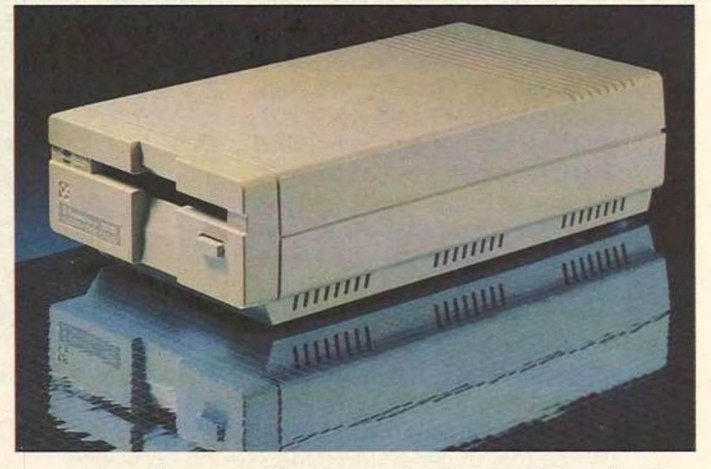
Hardware-Test

Listings

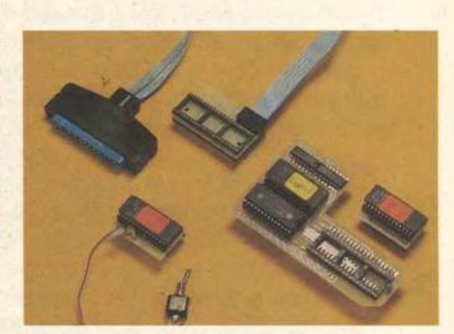
Tips & Tricks



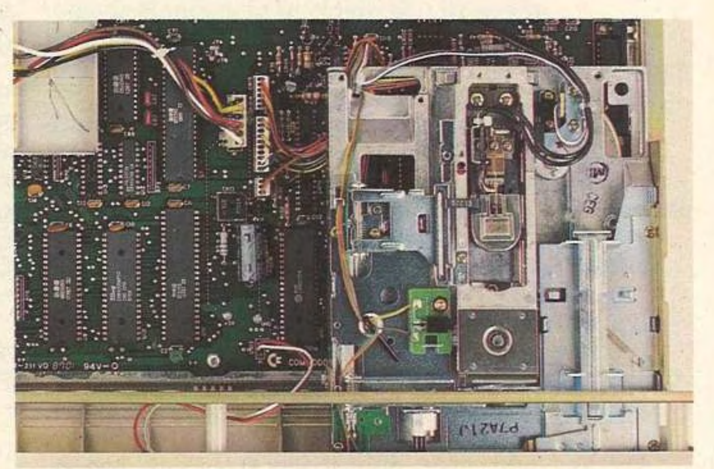
Die Floppy 1541c wird schon eine Weile verkauft. Alle Erfahrungen zeigen, daß sie nicht voll kompatibel zur 1541 ist. Was ist anders? **Seite 10**



Ein neues Diskettenlaufwerk ist auf dem Markt – Commodore 1581. Ein 3½-Zoll-Laufwerk, das die Leistungen der Floppy 1571 noch übertrifft. Wir haben es genau unter die Lupe genommen. **Seite 6**



»Floppy-Speeder« – ein Thema, das immer von Interesse ist, weil ständig eine Verbesserung oder etwas Neues entwickelt wird. **Seite 18**

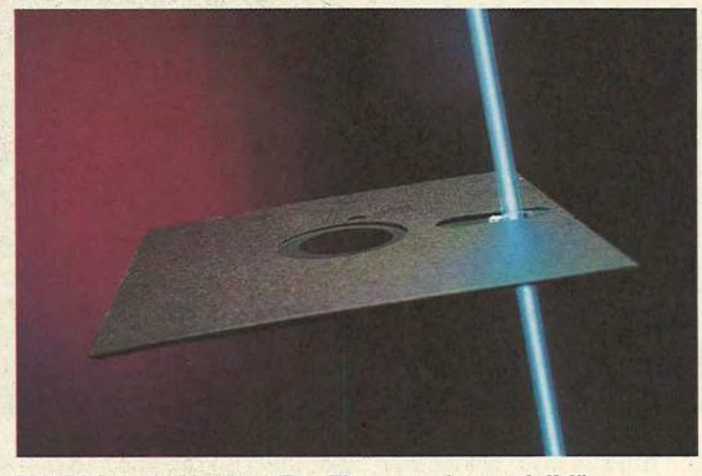


Auf dem neuen C128D mit Blechgehäuse laufen etliche Programme nicht mehr. Auch mit dem Einbau von Floppy-Speedern ergeben sich Probleme. Wir decken die Ursachen dafür auf. **Seite 16**





»XDOS« – ein umfangreiches Utility für den Umgang mit 1541-Disketten, das auch Ihre Wünsche befriedigen wird. **Seite 51**



Der Disketten-Monitor »Top-Flop« wurde speziell für die Floppy 1571 entwickelt, erkennt aber auch 1541-Formate. Sogar CP/M-Disketten für die 1571 können Sie beidseitig untersuchen. Seite 28



Wie Sie den Prozessor in der Floppy 1541 programmieren und für selbständig ablaufende Prozesse nutzen können, erfahren Sie ab Seite 117

```
*** MASTER - COPY PARALLEL V 1.3 ***
(C) 1987 BY FRANK RIEMENSCHNEIDER

ANZAHL KOPIEN: 01

* MASTER-COPY PARALLEL V 1.3 *
PROGRAMMENTWICKLUNG 1/87-5/87
FRANK RIEMENSCHNEIDER
GEORG-BUECHNER-STR. 5 - 3 H 71
KOPIERZEIT 40 TRACKS: 18 SEK.
PRUEFZEIT 40 TRACKS: 9 SEK.
UEBERTRAGUNGSRATE: 27,5 KB/SEK

STARTTRACK: 01
ENDTRACK: 40
GERAETENUMMER: 08
PROGRAMMENDE
INFO
```

»Master-Copy parallel« – ein superschnelles Kopierprogramm für die Floppy 1541 mit Parallelkabel. In 18 Sekunden wird eine komplette Diskette mit bis zu 40 Spuren kopiert und überprüft. Seite 44

Grundlagen

Überblick

Eingabehilfen

Sonstiges

Alle Programme aus Artikeln mit einem ■ -Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft



3½-Zoll: Die neue Floppy 1581

Endlich ist sie erhältlich, die Commodore 1581. Ein neues Diskettenlaufwerk für den C128 mit vielen Vorteilen. Vom 3½-Zoll-Diskettenformat bis hin zu 1 MByte Speicherkapazität.

Schon vor längerer Zeit angekündigt, ist sie jetzt endlich auf dem Markt, die 1581 (Bild 1). Hierbei handelt es sich um ein Laufwerk, das mit den mittlerweile zum Standard gewordenen 3½-Zoll-Disketten arbeitet, wobei natürlich neben den Vorteilen dieser Disketten (Kompaktheit, Stabilität, höhere Datensicherheit) auch sofort der erste Nachteil der 1581 ins Spiel kommt: Alle Programme, die speziell auf das 5¼-Zoll-Format zugeschnitten sind, arbeiten prinzipiell nicht mit der 1581 zusammen. Professionelle Programme wie Textverarbeitungssysteme oder Datenbanken funktionieren jedoch mit der 1581 und können deren Vorteile voll nutzen.

Da es sich bei der 1581 um ein sehr kleines und kompaktes Laufwerk handelt, hat Commodore das Netzteil wie beim C128 aus dem Gehäuse ausgelagert. Sie bekommen auf diese Weise zwar mehr »Kabelsalat« am Arbeitsplatz, dafür gibt es keine thermischen Probleme beim Dauereinsatz der Floppystation. Mit dem größeren »Kabelsalat« und den Problemen bei der speziellen Software für 5¼-Zoll-Formate sind die Nachteile der 1581 auch schon alle aufgezählt.

Unterschiede zur 1571

Zu den vielen Vorteilen, die das neue Laufwerk bietet, zählt an erster Stelle die riesige Speicherkapazität von 1 MByte pro Diskette. Für den Anwender heißt das: Etwa 800 KByte sind für Programme und Daten frei verfügbar! Das spricht sicherlich diejenigen an, die den Computer (die 1581 wurde speziell für den C128 entwickelt) in erster Linie professionell einsetzen wollen. Wir können Ihnen schon jetzt sagen, daß Sie mit der neuen Floppystation voll auf Ihre Kosten kommen; da verblaßt sogar die 1571.

An erster Stelle sei gesagt: Generell ist die 1581 kompatibel zur 1571. Sie enthält zumindest im großen und ganzen den gleichen Befehlssatz, doch dazu später mehr. Im Gegensatz zur 1571 hat Commodore in die 1581 ein paar zusätzliche Hardware-Voraussetzungen eingebaut, die einen schnelleren und fehlerfreieren Betrieb garantieren.

Zu diesen Voraussetzungen zählt zuerst das Format der Diskette. 3½-Zoll-Disketten sind – es wurde schon darauf hingewiesen – kleiner, stabiler und verfügen demzufolge über eine höhere Datensicherheit. Außerdem sind die Laufwerke schneller in der Datenverarbeitung und verbrauchen weniger Strom.

Eine weitere Voraussetzung für schnelleren Betrieb ist die höhere Speicherkapazität der Disketten. Häufige Diskettenwechsel können Sie also erst einmal vergessen.

Neben den Disketten wurde auch die Floppystation mit einer höheren internen Speicherkapazität ausgestattet. Im Gegensatz zur 1571 (2 KByte) besitzt die 1581 8 KByte Pufferspeicher. Inwieweit sich das auf die Verarbeitungsgeschwindigkeit auswirkt, soll später besprochen werden. An dieser Stelle sei jedoch schon soviel gesagt: Die 1581 arbeitet mit dem Prinzip eines »Caché-Speichers« (»Caché« kommt aus dem Französischen und wird demzufolge nicht englisch »Käsch«, sondern »Kaschee« ausgesprochen!). Dieses Speicherprinzip ermöglicht eine höhere Arbeitsgeschwindigkeit, indem immer eine komplette Spur von der Diskette in den internen Puffer übernommen wird (später mehr dazu).

Arbeitete die 1571 noch mit zwei verschiedenen Diskettenformaten (GCR und MFM), so wurde bei der 1581 komplett auf das GCR-Format verzichtet. Für den Anwender macht das keinen Unterschied; der floppyinterne Verwaltungsaufwand schrumpft jedoch auf ein Minimum. Für die technisch weniger versierten Leser an dieser Stelle ein Hinweis: Das GCR-Format verwendet die 1571 immer dann, wenn mit dem Computer im C128- oder C64-Modus gearbeitet wird und auch im CP/M-Modus unter Commodore-Format. Bei den Fremdformaten anderer Computerhersteller, die die 1571 jedoch zusätzlich bearbeiten kann, handelt es sich ausschließlich um MFM-Formate. Das ist eine andere Form der magnetischen Aufzeichnung auf Diskette; die-

sen Unterschied bemerkt der Anwender jedoch nicht. Er macht sich erst bei der fortgeschrittenen Laufwerksprogrammierung bemerkbar. Auch auf das Diskettenformat kommen wir später noch einmal zurück.

Die Hardware

Zuerst soll uns das Innenleben der 1581 interessieren. Schrauben wir sie also auf (Bild 2 und 3). Die Elektronik erkennt man erst, nachdem das Laufwerk entfernt wurde. Wie Sie sehen, sind die wichtigen Bauteile auf der Platine bezeichnet (Bild 3). Dabei handelt es sich um den Mikroprozessor 6502A (5), ein RAM 4364 mit 8 KByte Speicherkapazität (4), das 32-KByte-EPROM mit dem DOS der 1581 (3), einen CIA 8520A (2) und den Diskcontroller WD 1772 (1). Für jeden, der die Floppy 1571 schon einmal offen gesehen hat, ist unschwer zu erkennen, daß der Aufbau der 1581 sehr viel einfacher und demzufolge auch preiswerter und weniger störanfällig ist.

Das Netzteil wird an der Rückseite mit einem Stecker angeschlossen. Die Buchsen für den Anschluß des seriellen Buskabels sind identisch zu denen der 1571; also zwei gleichwertige 6polige DIN-Buchsen. Zwei von außen zugängliche DIP-Schalter erlauben das Umstellen der Geräteadresse der 1581 zwischen 8 und 11.

Die Software

Neben der Hardware mußte natürlich auch ein komplett neues Betriebssystem entwickelt werden, das in der Lage ist, die große Speicherkapazität und die vielen Funktionen der 1581 zu verwalten. Wie schon erwähnt, wurden nahezu alle Funktionen der 1571 übernommen, um eine größtmögliche Kompatibilität zu erreichen. Darüber hinaus sind einige gravierende Änderungen und Erweiterungen erfolgt, die wir uns nun betrachten wollen.

Es existieren nach wie vor die Befehle FORMAT, LOAD, SAVE, SAVE mit »@«, VERIFY, SCRATCH, RENAME, COPY, VALIDATE und INITIALIZE. Beim Speichern mit dem Klammeraffen »@« können wir jedoch noch nicht sagen, ob der verhängnisvolle Betriebssystemfehler, der im DOS der 1541, 1541c, 1570 und 1571 vorhanden war, beseitigt wurde. Wir können also nur an Sie appellieren, vorsichtig mit dem Ersetzen von Dateien umzugehen. Sollten Sie jedoch einmal dadurch auf dubiose Weise Daten verlieren, dann teilen Sie uns das bitte mit, damit wir diese Information auch vielen anderen Anwendern zugänglich machen können.

Als Dateitypen stehen bei der 1581 alle schon bekannten zur Verfügung, PRG, SEQ, USR und REL-Dateien. Auch an der Bedienung dieser Filetypen hat sich nichts geändert, so daß wir dieses Kapitel überspringen können.

Bei den floppyinternen Befehlen wird es jedoch interessant. Neben den üblichen Kommandos, wie BLOCK-READ, BLOCK-WRITE, MEMORY-EXECUTE, und so weiter, wurden auch neue, zusätzliche Befehle eingeführt. Einer dieser Befehle ist das PARTITION-Kommando. Das Wort »Partition« kommt aus dem Englischen und steht für »Partitionierung« oder »Unterteilung«. Vielleicht haben Sie schon

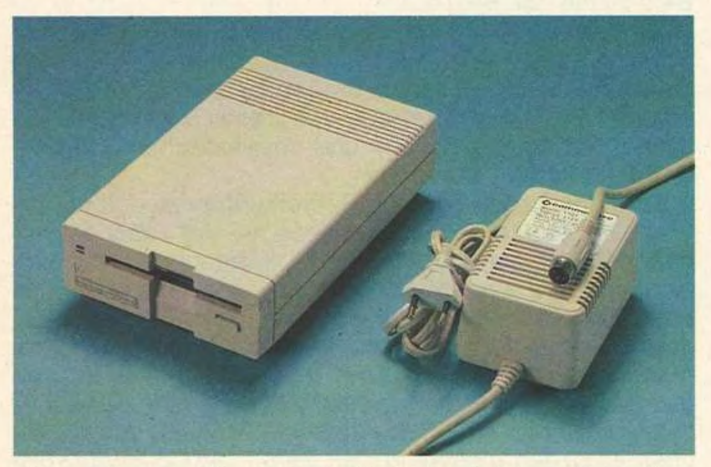


Bild 1. Die neue Floppy 1581 mit dazugehörigem Netzteil

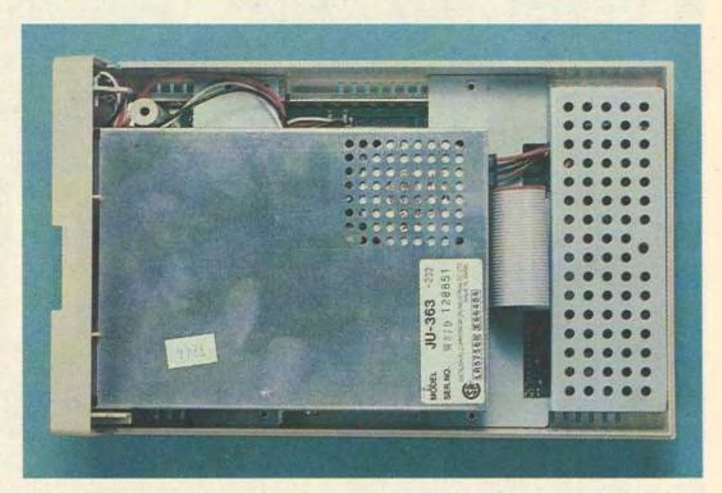


Bild 2. Die 1581, geöffnet von oben. Die Elektronik wird von dem $3^1/_2$-Zoll-Laufwerk vollkommen verdeckt.

einmal etwas von Unter- oder Subdirectories gehört? Hierbei handelt es sich um Directories, die in einem herkömmlichen Directory oder Inhaltsverzeichnis stehen. Das hat den Vorteil, daß man beim Ansehen eines Disketteninhalts nicht eine Unmenge von Programmeinträgen zu Gesicht bekommt, sondern nur einige wenige, nach Sachgebieten aufgeteilte Unter-Directories. Möchte man zum Beispiel ein spezielles Programm laden, das im Unterinhaltsverzeichnis »Diskmonitore« steht, dann wählt man dieses Subdirectory an. LISTet man es, dann erkennt man die einzelnen Programme, die zu dieser Rubrik gehören und kann sich das gewünschte aussuchen.

Die Floppy 1581 ermöglicht das Anlegen von Subdirectories mit dem PARTITION-Befehl, wobei der Anwender dieser Partition sogar einen speziellen Teil der Diskette als reservierten Platz zuweisen kann. Im Haupt-Directory erscheint eine Partition mit dem gegebenen Namen und dem Dateityp »CBM«.

Caché-Speicher und Burst-Kommandos

Eine spezielle Eigenschaft der 1571 gegenüber der 1541 ist der schnelle serielle Bus. In dieser, als Burst-Modus bezeichneten, Betriebsart können Datenbits mit einer Übertragungsrate von bis zu 1 Million Bit/s gesendet und empfangen werden. Zum Vergleich: der langsame serielle Bus der 1541 arbeitet mit 2400 Bit/s.

Auch die 1581 beinhaltet die Burst-Routinen und die entsprechenden Kommandos. Sie arbeitet jedoch um einiges schneller als die 1571, weil Commodore ein paar Besonderheiten eingebaut hat. Eine davon ist der schon erwähnte Caché-Speicher. Der Ausdruck hat sich mittlerweile derart eingebürgert, daß der »Accent aigu« (gemeint ist der nach rechts geneigte »Haken« über dem e) bei »Caché« häufig weggelassen wird. Im Englischen schreibt man deshalb auch oft »cache memory« oder »cache buffer«. Wir wollen jedoch die Original-Schreibweise beibehalten.

Normalerweise arbeitet ein Diskettenlaufwerk folgendermaßen: Wird ein Programm geladen, das aus mehreren Sek-

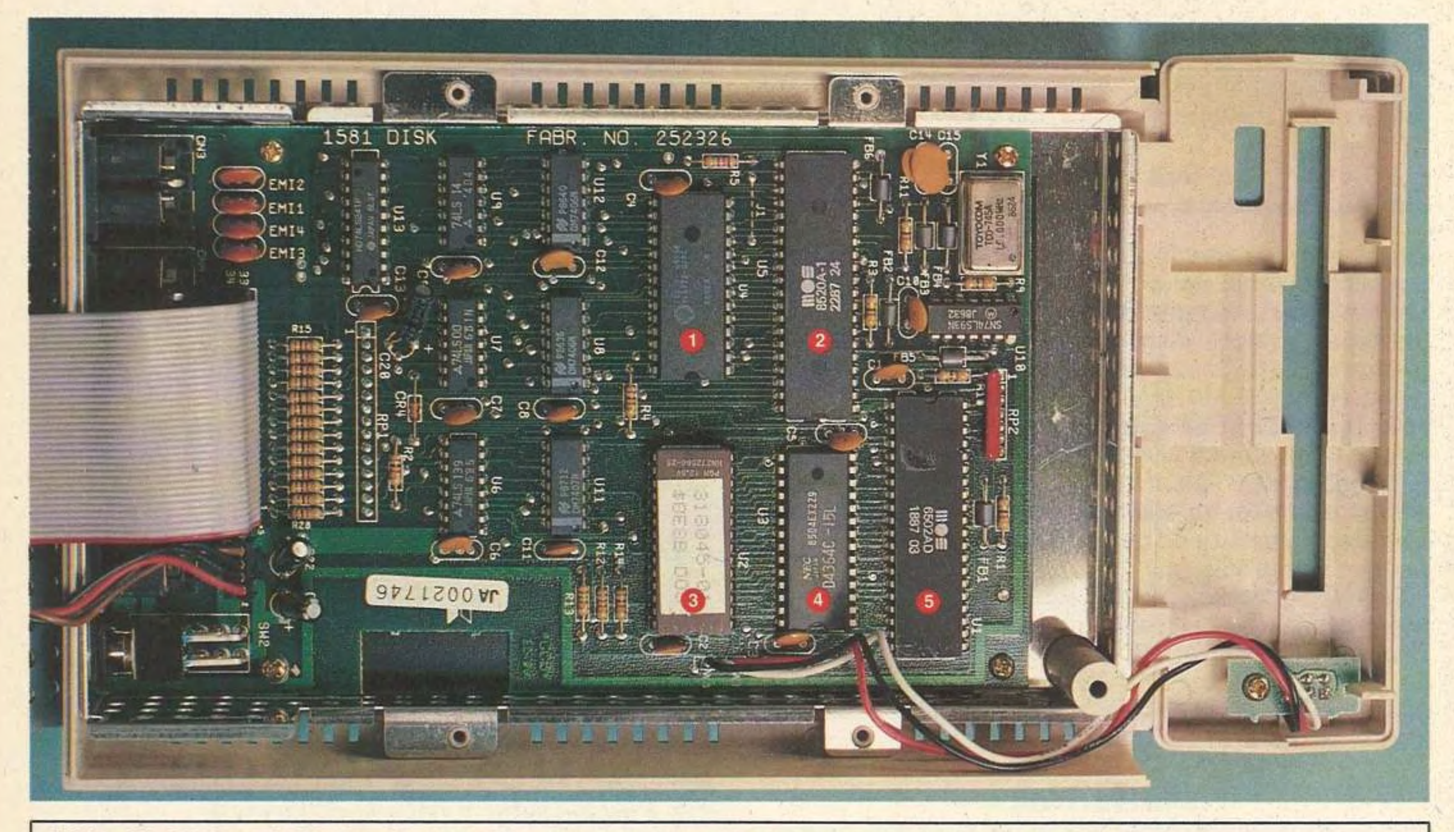


Bild 3. Die Elektronik-Platine der 1581. Die einzelnen Bauteile sind gekennzeichnet, und zwar:
1 – Diskcontroller WD 1772 für die Laufwerkssteuerung
2 – CIA 8520A für die Bedienung des seriellen Bus und des Laufwerks
3 – EPROM mit dem 32-KByte-Betriebssystem der 1581
4 – RAM 4364 mit 8 KByte Kapazität
5 – Mikroprozessor 65002A

toren besteht, so wird zunächst der erste der Sektoren von der Diskette in einen floppyinternen Zwischenspeicher (Puffer) geladen; danach erfolgt die Übertragung zum Computer. Ist das erledigt, wird der nächste Sektor auf der Diskette gesucht, geladen und so weiter. Dieses Prinzip ist zwar sehr einfach, kostet jedoch auch unnötige Zeit. Immerhin muß das Diskettenlaufwerk nach jedem übertragenen Datenblock erst den folgenden Sektor auf der Diskette suchen. Während dieser Zeit wartet der Computer »nichtstuend« auf weitere Daten.

Die 1581 arbeitet etwas anders und spart dadurch viel wertvolle Zeit ein. Sie liest immer eine komplette Spur und nicht nur einen einzelnen Sektor in ihren riesigen Puffer – eben jenen Caché-Speicher. Dadurch kann sie doppelt bis dreimal so schnell arbeiten wie die ohnehin schon relativ schnelle 1571. Im Gegensatz zur 1541 bedeutet das eine Geschwindigkeitserhöhung um das bis zu 30fache!

Da die Floppy 1581 nur ein Diskettenformat verwendet, nämlich das MFM-Format, entfällt eine weitere, sehr nervenaufreibende Eigenschaft der 1571: die Formaterkennung. Vielleicht haben auch Sie schon mit einer 1571 gearbeitet, dann kennen Sie folgendes Phänomen: Nach dem Einlegen einer Diskette im 1541-Format benötigt die 1571 eine ewige Zeit, bis sie beispielsweise das Directory oder ein spezielles Programm findet. Währenddessen flackert die grüne LED am Laufwerk hektisch und der Computer meldet lediglich »SEARCHING FOR...«. Die 1581 ist da um einiges schneller, da sie erstens keinen 1541-Modus kennt (wozu auch, bei 3½-Zoll-Disketten) und natürlich auch die Directoryspur in ihrem Caché-Speicher verwaltet.

Da wir uns schon über Spuren und Sektoren unterhalten, ist es angebracht, einmal über den Aufbau einer Diskette zu sprechen, nachdem diese mit der 1581 formatiert wurde. Dazu zum Vergleich die Daten der 1571: Sie arbeitet normalerweise zweiseitig, wobei zuerst die Seite 0 mit 35 Spuren zu je 17 bis 21 Sektoren vollgeschrieben wird, wobei jeder Sektor 256 Datenbytes enthält. Anschließend erfolgt das Umschalten auf die Seite 1, die ebenfalls über 35 Spuren zu je 17 bis 21 Sektoren verfügt. Für die variable Sektorzahl ist dabei der abnehmende Durchmesser der Spuren von außen nach innen verantwortlich.

Die 1581 organisiert ihre Disketten prinzipiell genauso wie die 1571; es erfolgt also eine Aufteilung in Spuren und Sektoren. Nur wird hierbei nicht zwischen Seite 0 und 1 unterschieden. Das heißt: Die 1581 besitzt 80 Spuren auf jeder Diskettenseite (also insgesamt 160); es werden jedoch die jeweils übereinanderliegenden Spuren zu einer Einheit zusammengefaßt. Diese bezeichnet man als »Zylinder«. Pro Zylinder verfügt die 1581 über 20 physikalische Sektoren (10 auf jeder Diskettenseite). Da diese Sektoren je 512 Datenbytes enthalten (1571: 256 Byte), teilt die Floppystation sie automatisch in je zwei logische Sektoren (oder Blöcke) mit 256 Byte auf.

Der Anwender bemerkt von sämtlichen organisatorischen Vorgängen nichts, so daß sich die 1581 für ihn wie ein einseitiges Laufwerk mit 80 Spuren zu je 40 Sektoren verhält; der Programmierer muß sich jedoch über den Unterschied zwischen logischen und physikalischen Sektoren und Zylindern im klaren sein.

Am Schluß noch ein paar Bemerkungen zu den floppyinternen Befehlen. Wer sich schon mit der Jobcode-Programmierung eines Commodore-Diskettenlaufwerks auseinandergesetzt hat, wird in der 1581 ein wahres Paradies entdecken können. Es stehen Ihnen nicht weniger als 33 Jobcodes zur Verfügung, die auch die exotischsten Programmierobjekte unterstützen. Vielleicht ein kleiner Vergleich: die 1571 verfügt gerade über neun Jobcodes.

Insgesamt ist die Floppy 1581 ein Produkt, das mit Sicherheit zu den ausgereiftesten gehört, die Commdore bisher vorgestellt hat. (Karsten Schramm/kn)



Floppy 1541c kontra Floppy 1541

Es gibt sie zweimal, die Floppy 1541: als »normale« 1541 und als neue 1541c. Worin liegen die Unterschiede der beiden Laufwerke, und welche Konsequenzen ergeben sich daraus?

Die Firmenstrategie von Commodore ist klar zu erkennen: Zum einen sollte die Floppy 1541 betriebssicherer werden; zum anderen wünschte man eine zusätzliche Einsparung von Produktionskosten. Das Ergebnis der Schönheitsoperation zeigte sich bald: Eine neue Version der bewährten 1541 entstand - die Floppy 1541c. Es stellt sich nun die Frage, was sich außer dem Namen noch alles geändert hat. Immerhin sind Hardware-Modifikationen nahezu zwangsläufig mit Kompatibilitäts-Problemen für die erhältliche Software (und das ist eine ganze Menge) verbunden. Die mögliche Folge: Es läuft nicht mehr alles.



Bild 1. Äußerlich kaum zu unterscheiden: die Floppy 1541 (links) und die 1541c (rechts). Während viele (nicht alle) ältere Modelle noch mit Klappverschluß versehen sind, besitzt die 1541c ein Mitsumi-Laufwerk mit Knebelverschluß.

Die Floppy 1541c gibt es schon eine Weile auf dem Markt, während die Produktion der »alten« 1541 (Bild 1, links) eingestellt wurde. Hard- und Software-Entwickler haben sich auf das neue Peripheriegerät (Bild 1, rechts) »eingeschossen«, und die Welt scheint in Ordnung. Aber der Schein trügt. An einem Heimcomputer wie dem C 64 wird zwangsläufig sehr viel herumgebastelt, und viele Tüftler machen auch vor der Floppystation nicht halt. Schneller muß sie werden, mehr LED-Anzeigen besitzen, zwischen Laufwerksnummer 8 und 9 umschalten können oder mehrere Betriebssysteme besitzen.

Probleme mit der Kompatibilität

Praktisch alle Eingriffe in die Hardware der 1541 und 1541c sind mit dem Öffnen des Gehäuses (Vorsicht: vorher Netzstecker ziehen!) verbunden. Und schon kommt die Überraschung bei der 1541c zutage (Bild 2). Wie Sie sehen, ist nicht nur die Anordnung der Bauteile modifiziert worden, es wurden auch einige alte Bausteine gegen höher integrierte ausgetauscht. Die beiden ROMs (4), die in der 1541 das Betriebssystem enthalten, wurden durch ein ROM (4a) mit der doppelten Speicherkapazität ersetzt. Statt einer Unmenge an passiven und aktiven Elektronikbauteilen zur Signalverstärkung bei Schreib- und Lesevorgängen (7) wurde in der 1541c ein Hybrid-Schaltkreis (7a) eingesetzt.

Wollen Sie zum Beispiel die Geräteadresse Ihrer 1541c umstellen, so müssen Sie die Lötbrücken an einer anderen Stelle als bei der 1541 suchen. Bild 3 zeigt Ihnen die Positionen im Vergleich (rechts die 1541c).

Das »Rattern« hat ein Ende

Neben den Eingriffen, die die 1541c nur elektronisch auf einen neueren Stand bringen, hat Commodore auch einen kleinen Zusatz eingebaut, der das von der 1541 her bekannte »Rattern« (zum Beispiel beim Formatieren einer Diskette) vermeiden hilft. Die Ursache für den lästigen Lärm, der nicht nur gefährlich klingt, sondern auch die Lebensdauer des Laufwerks verringert, ist folgende Eigenschaft der 1541: Da die Floppystation die Position ihres Schreib-/Lesekopfes nicht absolut, sondern nur relativ zu einer anderen Position bestimmen kann, benötigt sie zumindest einmal nach dem Einschalten einen Fixpunkt, von dem aus alle weiteren Positionen berechnet werden können. Dieser Fixpunkt ist die Spur 0. (Wie Sie vielleicht wissen, arbeitet die 1541 üblicherweise mit den Spuren 1 bis 35.) Um den Fixpunkt zu erreichen, läßt die 1541 den Schreib-/Lesekopf, egal wo er sich befindet, 44 Spuren in Richtung Spur 0 zurückfahren. Nehmen wir einmal an, der Kopf steht auf der Spur 18; die Floppystation »weiß« das aber nicht. Jetzt beginnt das Zurückfahren. Nach dem Überspringen der 18 Spuren kommt der Schreib-/Lesekopf auf Spur 0, an der sich eine genau eingestellte, mechanische Barriere befindet. Die restlichen 25 Spurwechsel kann sich der Schlitten nicht mehr weiterbewegen, mit dem Erfolg, daß er 25mal gegen die Barriere schlägt. Nicht sehr schonend für die Mechanik, aber wirkungsvoll: selbst wenn sich der Schreib-Lese-Kopf auf einer undefinierten Spur von beispielsweise 39 befindet, wird er durch die eben beschriebene Methode ganz sicher auf Spur 0 »verfrachtet«, um dann neu positioniert zu werden.

Allerdings schlägt der Schreib-Lese-Kopf bei jedem Formatiervorgang und bei Lesefehlern an, was eine erhebliche mechanische Belastung bedeutet.

Die Spur-Null-Erkennung kann man aber auch auf eine elegantere (und schonendere) Weise realisieren. Bei der 1541c existiert an der Spur-Null-Position eine Lichtschranke, die die Annäherung des Schreib-Lese-Kopfs »bemerkt«, und sofort stoppt der Motor die Kopfbewegung - noch bevor ein mechanischer Anschlag erfolgt. Es ist schon verblüffend, wie leise ein Diskettenlaufwerk arbeiten kann.

Der Einbau dieser zusätzlichen Lichtschranke schont zwar das Laufwerk erheblich, stellt sich aber gleichzeitig als ein großes Problem bei der 1541c heraus. Diverse Soft- und Hardware, die für die 1541 entwickelt wurde, weigerte sich standhaft, mit der 1541c zusammenzuarbeiten. Die Gründe für diese Inkompatibilität zwischen den beiden Floppystationen sind folgende:

Der erste Punkt betrifft die Software der 1541c, das Betriebssystem (DOS). Damit die Spur-Null-Abfrage über die Lichtschranke überhaupt möglich wird, ist eine Änderung der Betriebssystem-Routinen zur Bewegung des Schreib-Lese-Kopfes notwendig. Diese dürfen den Kopf nicht mehr 44 Spuren zurücktransportieren, sondern müssen statt dessen zusätzlich eine Abfrage der Lichtschranke durchführen. Da jedoch die 1541 bekanntermaßen recht langsam arbeitet, haben es sich viele Programmierer zu eigen ge-

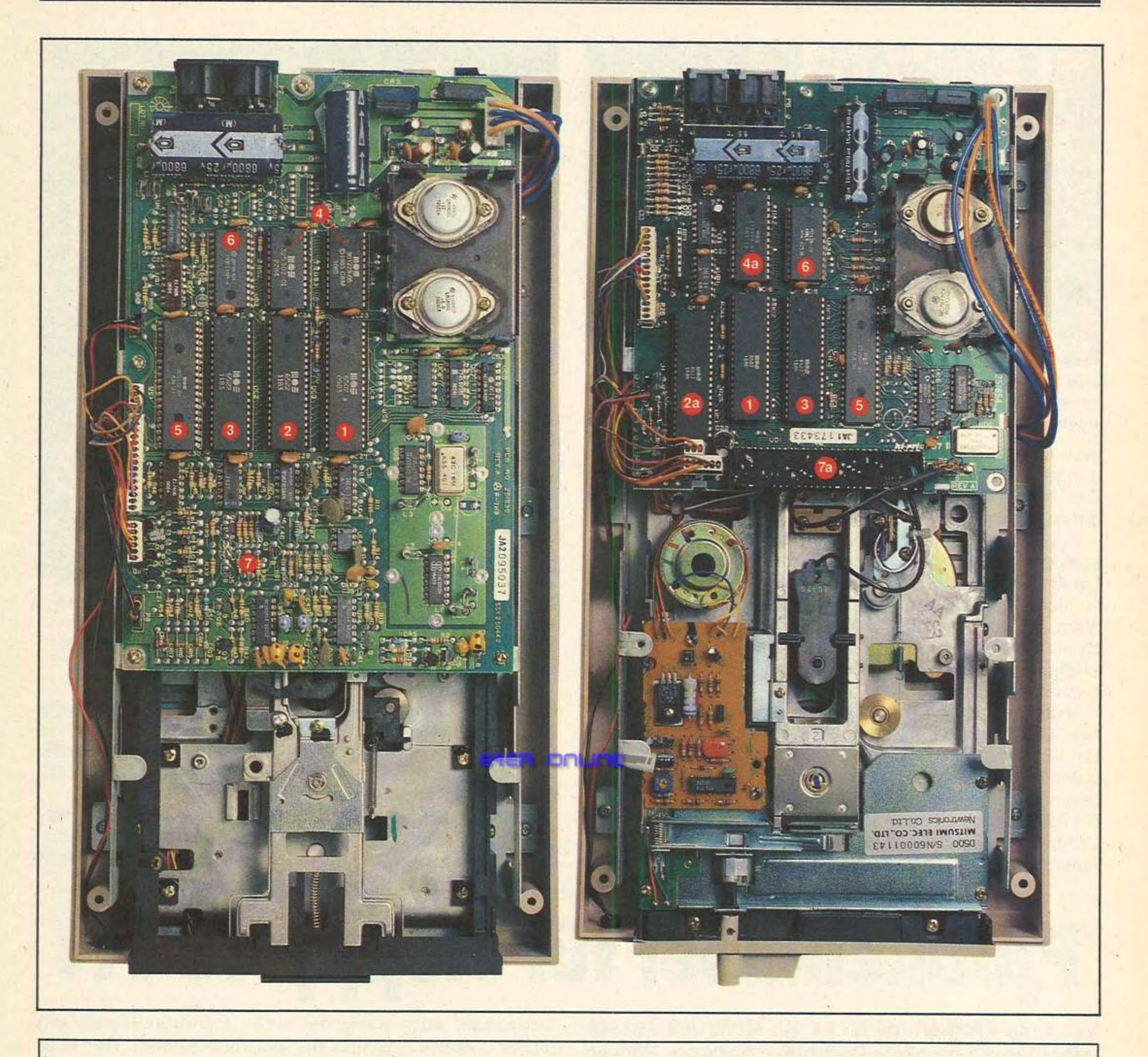


Bild 2. Erst nach dem Aufschrauben und Abnehmen des Gehäusedeckels werden die Unterschiede zwischen der 1541 (links) und der 1541c (rechts) deutlich. Die Platine mit der gesamten Steuerelektronik der 1541c ist »geschrumpft« und bedeckt nur noch den hinteren Teil des Laufwerks und den darunterliegenden Transformator. Die wichtigen Bauteile:

1 – Mikroprozessor 6502 (1541 und 1541c)
2 – CIA 6522 1 für die Bedienung des seriellen Bus (nur 1541)
2a – CIA 6522 1 für die Bedienung des seriellen Bus und die Abfrage der Spur-Null-Lichtschranke (nur 1541c)
3 – CIA 6522 2 für die Datenübertragung zwischen Laufwerk und Elektronik (1541 und 1541c)
4 – zwei 8-KByte-ROMs (pinkompatibel mit EPROMs vom Typ 2564) für das DOS (nur 1541)
4a – ein 16-KByte-ROM (pinkompatibel mit EPROMs vom Typ 27128) für das DOS (nur 1541c)
5 – Disk-Controller für das Laufwerk der Floppystation (1541 und 1541c)
6 – 2-KByte-RAM (1541 und 1541c)
7 – Analogteil der Platine mit der Elektronik zur Verstärkung der Signalströme vom und zum Schreib-Lese-Kopf (nur 1541)
7a – Hybridbaustein; verantwortlich für die Signalverstärkung (nur 1541c).

macht, auch die Schrittmotorsteuerung für die Kopfbewegung neu zu programmieren und in ihre Programme einzubinden. Diese oft älteren Programme nehmen wenig Rücksicht auf Standardeinsprünge (Sprungtabellen) für die Routinen des DOS und kommen deshalb sehr schnell mit dem neuen Betriebssystem der 1541c in Konflikt. Ein zweiter Grund für die Inkompatibilität der 1541c zur 1541 ist die geänderte Hardware. In beiden Floppystationen übernimmt ein Ein-/Ausgabe-Baustein vom Typ CIA 6522 die Steuerung des seriellen Bus. Da dieser Baustein über 2*8 (also 16) Ein-/Ausgabe-Leitungen verfügt, vom seriellen Bus jedoch nur acht dieser Leitungen benötigt werden, liegen die restlichen Anschlüsse dieses Bausteins bei der 1541 brach. Hardware-Hersteller haben diese Eigenschaft

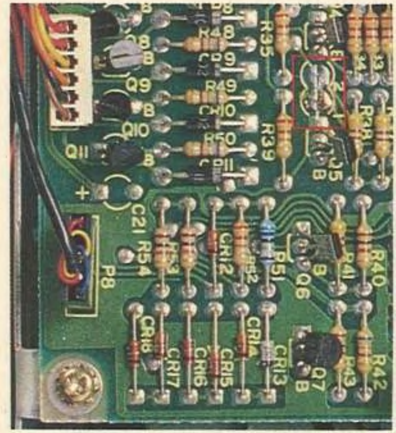
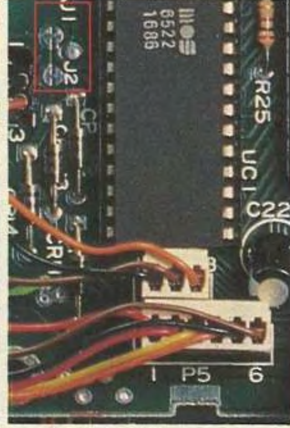



Bild 3. Das dauerhafte Umstellen der Geräteadresse geschieht bei beiden Floppystationen gleich; lediglich die Positionen der einzelnen Lötbrücken haben sich geändert

genutzt, um ohne großen Aufwand zusätzliche Übertragungsmöglichkeiten für Floppy-Speeder bereitzustellen. Die bekannten Parallelkabel der meisten Beschleunigungssysteme greifen auf die acht Leitungen zu.

Bei der 1541c wurde jedoch einer dieser acht »freien« Anschlüsse neu belegt. Er enthält die Information »Spur-Null-Lichtschranke unterbrochen« oder »Spur-Null-Lichtschranke nicht unterbrochen«. Dadurch ergibt sich aber ein Problem. Die parallele Datenübertragung zum C 64 kann künftig nicht mehr über die bisher unbenutzten acht Leitungen erfolgen. Wie es dennoch funktioniert, das verraten wir Ihnen im folgenden Artikel.

Vielleicht haben Sie sich gerade eine neue 1541c zugelegt. Dann werden Sie unter Umständen überrascht sein: Die Floppystation rattert wie jede 1541 – und das nicht nur beim Formatieren und bei Lesefehlern, sondern auch jedesmal nach einem Reset, beziehungsweise nach dem Einschalten. Der Grund hierfür liegt in der Tatsache, daß Commodore die Spur-Null-Lichtschranke bei etlichen neueren Geräten »außer Gefecht« gesetzt hat. Bild 4 zeigt

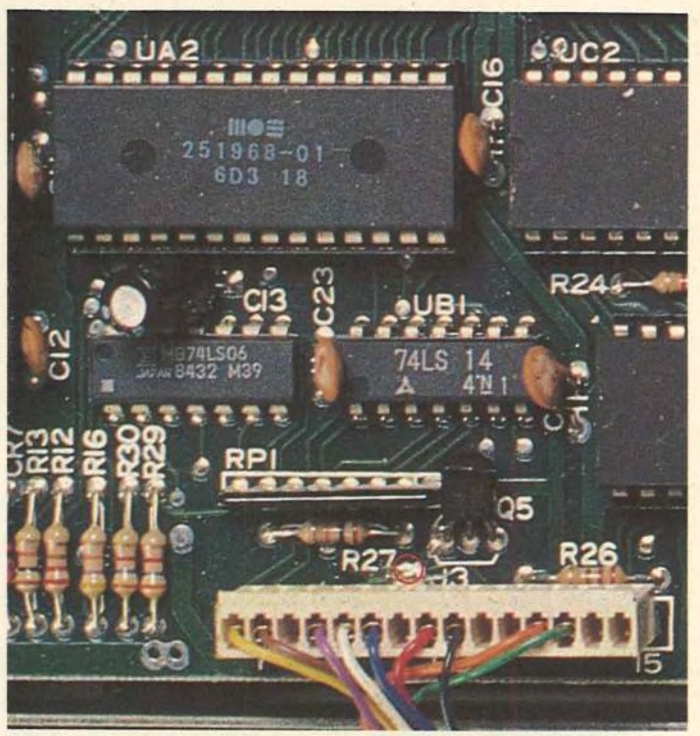


Bild 4. Wenn Sie die dargestellte Lötbrücke (auf der 1541c-Platine halb unter dem Stecker) durchtrennen, wird die Spur-Null-Position des Schreib-Lese-Kopfes mit Hilfe einer Lichtschranke ermittelt.

Ihnen eine Lötbrücke auf der Platine der 1541c. Ist diese Lötbrücke kurzgeschlossen, so ist die Lichtschranke inaktiv. Ein Öffnen dieser leitenden Verbindung schafft jedoch Abhilfe (Vorsicht: Unter Umständen Garantieverlust!). Das gefürchtete Rattern unterbleibt künftig.

Bis auf die beschriebenen Unterschiede sind keine weiteren Kompatibilitäts-Probleme zwischen der 1541 und der 1541c bekannt. (Karsten Schramm/kn)

Parallelkabel für Floppy 1541c

Für viele Programme ist es ein Muß: das Zusatzkabel für die parallele Datenübertragung. Was ist beim Einbau in die 1541c zu beachten?

Weit verbreitete Soft- und Hardware wird schnell zu einem Standard, wenn das Interesse unter den Computeranwendern groß genug ist. So war es auch bei dem Floppy-Speeder »Speeddos«. Dieses System hat sich auf dem Markt dank seiner einfachen Hardware und seiner großen Effizienz schnell durchgesetzt und ist schon seit langem der am weitesten verbreitete Beschleuniger für die 1541. Eine der Besonderheiten von Speeddos ist ein 10adriges Zusatzkabel, das für eine schnelle Datenübertragung zwischen Floppystation und Computer sorgt.

Einige Programme (zum Beispiel Kopierprogramme) haben sich mittlerweile die große Verbreitung von Speeddos zunutze gemacht und wurden speziell für das zusätzliche Parallelkabel geschrieben. Damit auch Sie mit solchen Programmen arbeiten können, beschreiben wir nun, wie ein solches Parallelkabel in eine 1541c eingebaut wird.

Generell ist das Kabel, das Sie für die 1541c benötigen, das gleiche, das auch in der 1541 Verwendung findet. Wir verzichten an dieser Stelle auf eine Bauanleitung für das Kabel, da Sie es für wenig Geld in nahezu jedem Computerladen oder auch direkt bei vielen Hardware-Herstellern (siehe unseren Anzeigenteil) beziehen können. Wer dennoch auf den Selbstbau Wert legt, der sei auf die Seite 44 dieser Ausgabe hingewiesen.

Zum Einbau schrauben Sie jetzt bitte das Gehäuse Ihrer 1541c (vier Schrauben an der Unterseite) auf und nehmen den Deckel ab. (Vorsicht: vorher unbedingt Netzstecker ziehen und die restlichen Kabelverbindungen zum Computer lösen!) Sie sehen jetzt vor sich das komplette Laufwerk mit der Platine für die elektronische Steuerung (Bild 1). Den auf unserem Foto mit der Nummer 1 markierten, 40poligen Baustein mit der Bezeichnung 6522 oder 65C22 hebeln Sie jetzt vorsichtig (!) mit einem kleinen Schraubenzieher aus seiner Fassung. Dabei beachten Sie bitte, daß integrierte Schaltkreise (ICs) sehr empfindlich gegenüber statischer Elektrizität sind. Sie sollten also sicherstellen, daß Sie nicht »aufgeladen« sind, indem Sie beispielsweise vor der Berührung eines ICs geerdete Metallteile (Wasserhahn, Heizung, elektrische Geräte mit Metallgehäuse) anfassen. Ist der Baustein jedoch ohne Sockel fest eingelötet, so wenden Sie sich am besten an einen Fachmann, der Ihnen das Bauteil ohne Zerstörung auslöten und sockeln kann.

Jetzt stecken Sie den 40poligen Baustein in den Sockel, der sich an dem einen Ende Ihres Parallelkabels befindet,

wobei Sie darauf achten müssen, daß das IC polrichtig (Kerbe auf Kerbe) in die Fassung eingepaßt wird. Bei der 1541 müßten Sie jetzt nur noch die Fassung mit dem IC polrichtig in den ursprünglichen Sockel des 6522 zu stecken und der Einbau wäre erledigt. Nicht so bei der 1541c. Hier ist zu beachten, daß dieses Laufwerk über eine zusätzlich eingebaute Lichtschranke zur Spur-Null-Erkennnung verfügt. Diese Lichtschranke muß leider »außer Gefecht gesetzt« werden, wenn das Parallelkabel funktionieren soll.

Die Lichtschranke ist an Pin 2 (das ist das zweite »Beinchen« links neben der Kerbe, von oben betrachtet) angeschlossen. Sie müssen nun dafür sorgen, daß dieser Anschluß noch Kontakt zu dem Sockel Ihres Parallelkabels hat, jedoch nicht mehr mit der Platine der 1541c in Verbindung steht. Kennen Sie sich in der Elektronik gut aus, dann ist es am besten, wenn Sie die zugehörige Leiterbahn auf der Platine unterbrechen. Für die weniger versierten Bastler hier eine zweite Möglichkeit: Kneifen Sie Pin 2 am Sockel des Parallelkabels mit einem Seitenschneider ab, wobei Sie jedoch aufpassen müssen, daß Sie den Anschluß des Kabels am Sockel nicht beschädigen. Jetzt geben Sie etwas Klebstoff über den Rest des abgetrennten Kontakts und warten bis dieser trocknet. Beim Einstecken in die Fassung auf der Platine darf Pin 2 keinen Kontakt mehr mit der Metallzunge der Fassung besitzen (Schemazeichnung in Bild 2).

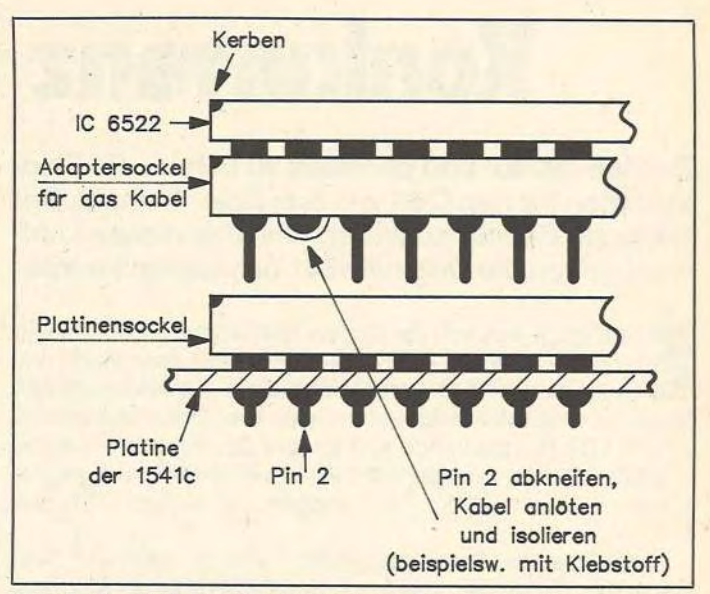


Bild 2. So muß die Verbindung des Pin 2 zwischen dem VIA 6522 und der Platine der 1541c unterbrochen werden.

Betriebssystem erneuern

Der letzte Schritt besteht noch im Austauschen des Betriebssystems. Das ist leider notwendig, da das DOS der 1541c die fehlende Lichtschranke nicht akzeptiert. Als Ersatz dient uns beispielsweise das Betriebssystem der »normalen« 1541. Sie benötigen jetzt einen EPROMer, ein leeres EPROM vom Typ 27128 und eine 1541. In der 1541 ist das Betriebssystem (DOS) in zwei ROMs untergebracht. Sie befinden sich in der hinteren IC-Reihe auf den Steckplätzen UB 3 und UB 4 (weißer Platinenaufdruck). Diese beiden ROMs hebeln Sie bitte vorsichtig aus den Fassungen und achten darauf, daß Sie sie nicht vertauschen. Starten Sie den EPROMer und stellen Sie als EPROM-Typ 2564 oder »Commodore-ROMs« ein. Wie das im einzelnen funktioniert entnehmen Sie bitte der Bedienungsanleitung des EPROMers. Stellen Sie jetzt die Startadresse im Computer auf $2000 (8192) und die Endadresse auf $4000 (16384), und lesen Sie das ROM von UB 3 mit dem EPROMer ein (Achtung: Die versehentliche Anwahl der Option »Programmieren« des EPROMers zerstört das ROM!!!). Direkt im Anschluß stellen Sie die Startadresse für das Einlesen im Computer auf $4000 (16384) und die Endadresse auf $6000 (24576). Anschließend lesen Sie das zweite ROM (aus Sockel UB 4) ein. Nehmen Sie das ROM aus dem Sockel des EPROMers und stellen Sie jetzt den EPROM-Typ 27128 ein. Normalerweise steht der Adreßbereich jetzt auf $2000 (8192) bis $5FFF (24575) oder $6000 (24576) und Sie können jetzt das EPROM programmieren.

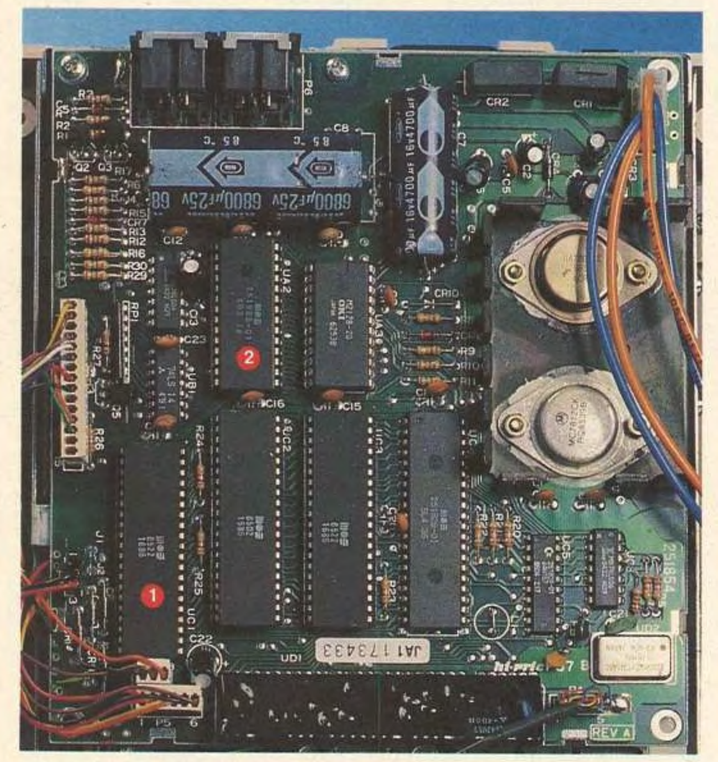


Bild 1. Die Platine der 1541c. Nummer 1 bezeichnet den Ein-/Ausgabe-Baustein VIA 6522, Nummer 2 das 16-KByte-ROM mit dem Betriebssystem der 1541c.

Bei manchen EPROMern ist dies nicht so einfach. Sie können dann aber beide Betriebssystem-ROMs zuerst speichern und anschließend mit einem Assembler-Monitor verketten zu einem brennfähigen File für das EPROM 27128. Wichtig ist, daß das ROM von UB 3 im EPROM 27128 ab Adresse $0000 liegt, während das ROM von UB4 bei $2000 (obere Hälfte im EPROM) beginnen muß.

Bei der 1541c hebeln Sie den Baustein aus dem Sockel, der in Bild 1 mit der Nummer 2 gekennzeichnet ist und setzen direkt das EPROM (polrichtig!) dafür ein.

Schrauben Sie jetzt das Gehäuse der 1541c wieder zu, wobei Sie das Flachbandkabel zwischen den beiden Gehäusehälften einklemmen. Stecken Sie den Netzstecker ein und schalten Sie die Floppystation probehalber, ohne angeschlossenen Computer, ein. Sie muß sich wie eine »normale« 1541 verhalten, das heißt kurzes Anlaufen des Laufwerkmotors, kurzes Aufleuchten der roten LED und anschließend Ruhe. Ist das nicht der Fall, läuft sie zum Beispiel bei eingeschalteter oder blinkender LED endlos weiter, dann haben Sie einen Fehler beim Einbau oder Brennen des EPROMs gemacht.

Funktioniert jedoch alles einwandfrei, dann schließen Sie das Laufwerk an den Computer an (im ausgeschalteten Zustand, versteht sich), und schalten Sie die komplette Computeranlage ein. Trotz angeschlossenem Zusatzkabel am Userport, muß alles wie gewohnt funktionieren. Lediglich Programme, die das Parallelkabel zusammen mit dem Standard-DOS der 1541 benötigen, laufen jetzt ebenfalls.

(Karsten Schramm/kn)

Konkurrenz für die 1541

Zierlich gebaut und preiswert zu haben: die Floppystation für den C 64 von Rex Datentechnik. Der folgende Bericht schildert, wie sich dieses Laufwerk gegen die Original-1541 behaupten konnte.

Es ist lange her, seit die letzten 1541-kompatiblen Diskettenlaufwerke von Fremdherstellern auf dem Markt waren, um der 1541 von Commodore die Vorherrschaft streitig zu machen. Jetzt versucht es Rex-Datentechnik mit einem Modell, das schon seit einiger Zeit für 329 Mark (ab 3 Stück 299 Mark) angeboten wird und bei dem einige gravierende Nachteile der 1541 ausgemerzt worden sein sollen.

Auffällig an der Rex-Floppy (Bild 1) sind die geringen Ausmaße des Gehäuses, die durch den Einsatz eines Slimline-Laufwerks mit einem externen Netzteil erkauft werden. Die Elektronik ist intern auf drei Platinen untergebracht, wobei

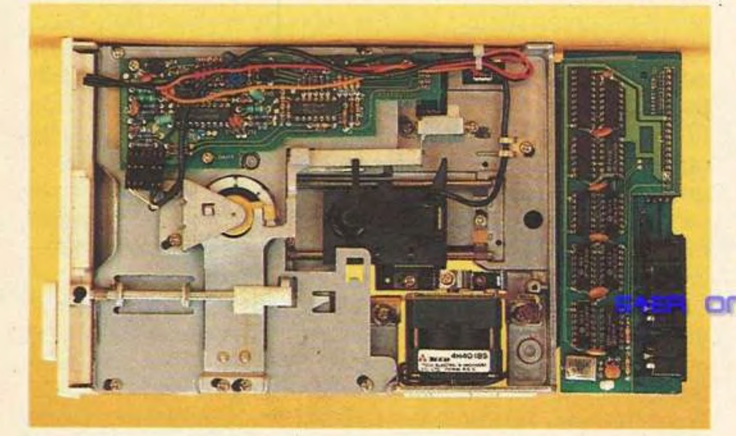


Bild 2. So sieht das Laufwerk geöffnet von oben aus . . .

die Digitalelektronik wegen Platzmangels auf der Unterseite des Laufwerks montiert wurde (Bild 2 und 3). Die Anordnung der drei 40poligen ICs (Mikroprozessor 6502 und zwei CIA 6522) ist identisch zur 1541; lediglich die beiden CIA 6522 wurden vertauscht.

Eine Steckkompatibilität zu vorhandenen Floppy-Speedern der neueren Generation (Dolphin-DOS, Turboaccess und Turbotrans) wurde dadurch leider zunichte gemacht, bis auf das seit kurzer Zeit von Rex Datentechnik vertriebene Prologic-DOS-Rex (Classic). Kompatibel ist das Laufwerk zu Speeddos und Rex-DOS. Ausgeliefert wird es serienmäßig mit eingebautem Rex-DOS-Betriebssystem. Der Anwender braucht sich also nur noch ein Parallelkabel besorgen, die Verbindung zum Computer herstellen und im Computer ein entsprechendes Rex-DOS-Kernel für 74,95 Mark einzusetzen. Der einzige Nachteil: Durch die unglückliche Anordnung der ICs in der Rex-Floppy kann das Gehäuse nach dem Einbau des Parallelkabels nicht mehr geschlossen werden (bzw. nur noch mit zwischengelegten 10-mm-Distanzrollen).

Dank Rex-DOS ist natürlich eine Geschwindigkeitssteigerung gegenüber der 1541 zu verzeichnen. Wird das Parallelkabel nicht montiert, so ist zumindest die Geschwindigkeit des Steppermotors erhöht, so daß eine schnellere Positionierung des Schreib-/Lesekopfes erfolgt. Das macht sich durch eine geringfügige Beschleunigung gegenüber der 1541 bemerkbar.

Wegen des externen Netzteils hält sich die Temperatur im Inneren des Diskettenlaufwerks stark in Grenzen. Thermische Probleme, wie sie teilweise bei der 1541 auftreten, sind somit nicht zu befürchten. Durch das Metallgehäuse (1541: Plastik) erfolgt darüber hinaus noch eine sehr gute Abschirmung des Laufwerks gegen fremde Störstrahlung.



Bild 1. Die Rex-Floppy »am Stück«

Das Laufwerk selbst machte im Test einen sehr guten Eindruck. Es ist solide, so daß auch das gefürchtete Anschlagen des Schreib-/Lesekopfes an der Spur-Null-Marke keinen Schaden anrichten kann. Das macht sich schon durch die sehr geringe Geräuschentwicklung bemerkbar. Während die 1541 ein ohrenbetäubendes Rattern von sich gibt, schnurrt die Rex-Floppy nur leise vor sich hin.

Der Befehlssatz des neuen Laufwerks ist identisch mit dem der 1541. Allerdings ist das Rex-DOS nicht 100prozentig kompatibel zum Original-DOS der 1541. Eine Auslieferung der Rex-Floppy mit dem Commodore-DOS ist aus urheberrechtlichen Gründen unmöglich.

Positiv an dem Gerät ist sein niedriger Preis, sein stabiles Metallgehäuse und das robuste Laufwerk. Auch der geringe Aufwand bei der Aufrüstung zum vollwertigen Rex-DOS-Laufwerk ist positiv.

Weniger gut ist hingegen die Tatsache, daß die Platinen im Gehäuse so ungünstig untergebracht sind und daß kein Platz für Erweiterungen bleibt. Für den Anwender, der oft

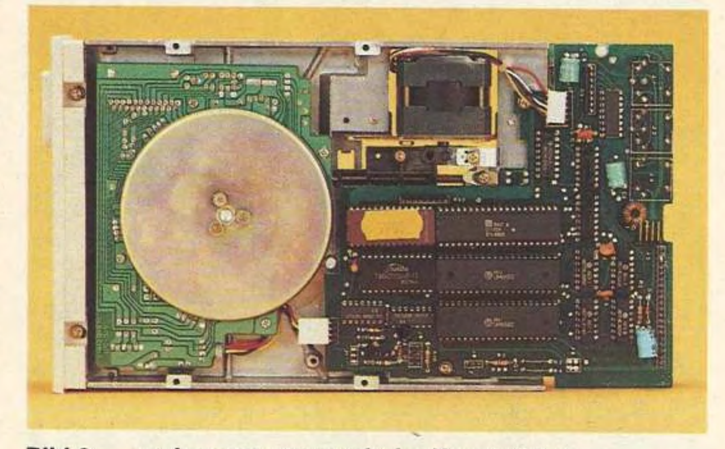


Bild 3. ...und so von unten mit der Hauptplatine

mit professioneller Software arbeitet, stellt sich weiterhin das Problem der Kompatibilität. Es laufen sicherlich nicht alle Programme mit dem Rex-DOS-Betriebssystem zusammen. (Karsten Schramm/kn)

Bezugsquelle: Rex Datentechnik, Stresemannstraße 11, 5800 Hagen 1, Telefon: 02331/32734

Halten Sie Ihre Floppy fest!

Zwei 64'er Sonderhefte bringen Power auf die Scheibe.



Das 64'er-Sonderheft 15

Alles Wissenswerte über Laufwerk und Datasette finden Sie in diesem Heft: Ausführliche Beiträge über Hardware- und Softwareprodukte, mit denen Sie alles aus Ihrem System herausholen. Verschiedene Kopierprogramme, die in Sachen Geschwindigkeit keine Wünsche mehr offen lassen.



Das 64'er-Sonderheft 9

Auf einen Blick: kompletter Floppy-Kurs aus den 64'er-Ausgaben. Über 60 Seiten Listings zum Abtippen aus den Bereichen Utilities, Tools, Tips&Tricks. Weitere Kurse erleichtern die eigene Dateiverwaltung durch Basic- und Assemblerprogrammierung. Im Test befinden sich die wichtigsten Floppy-Speeder.

Nutzen Sie die Bestellmöglichkeiten für diese zwei »64'er«-Sonderhefte mit der eingehefteten Zahlkarte in diesem Sonderheft oder fragen Sie Ihren Zeitschriftenhändler.

Ein Problemkind: die neue Floppy im Blechgewand

Seit einiger Zeit wird der C128D in einem neuen Blechgehäuse ausgeliefert. Seitdem häufen sich die Probleme mit dem eingebauten Laufwerk. Was ist anders bei diesem Computer? Wie läßt sich Abhilfe schaffen?

In der 64'er-Ausgabe 8/87 haben wir den neuen C128D erstmals vorgestellt. Die auffälligste äußere Veränderung ist das neue Blechkleid. Wir wollen die neue Version daher »C128D-Blech« nennen.

Nach dem Einschalten dieses Computers bleibt es gegenüber seinem Vorgänger angenehm ruhig - Commodore hat auf den Lüfter verzichtet. Eine positive Entscheidung. Weit weniger angenehm ist die Erfahrung, daß diverse Programme bei diesem Computer ihre Dienste versagen. Besonders Software, die genau auf die Floppy 1571 abgestimmt ist (wie beispielsweise das schnelle Kopierprogramm »Double-Touch« aus dem 64'er-Sonderheft 22), funktioniert auf dem neuen C128D-Blech überhaupt nicht. Woran liegt das?

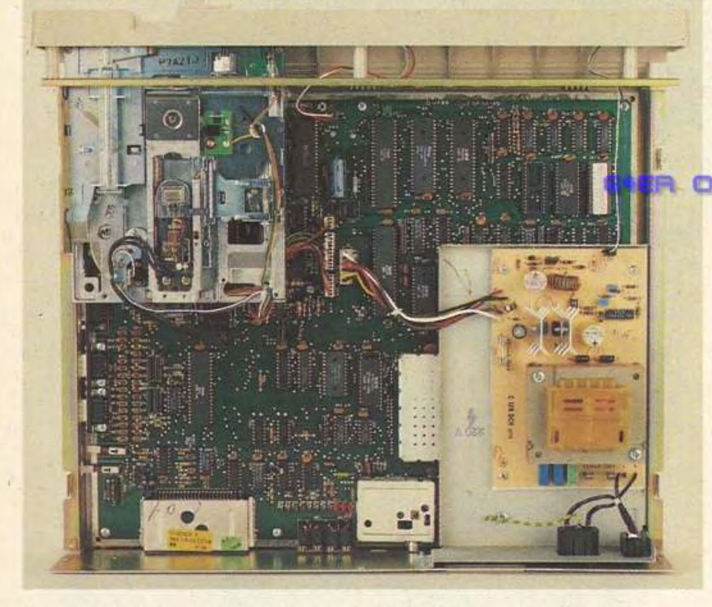


Bild 1. Der neue C128D mit Blechgehäuse. Die Floppy-Steuerung ist jetzt auf der Hauptplatine integriert.

Neue Platine

Öffnet man das Gehäuse, so zeigt sich ein neues Bild des Innenlebens (Bild 1). Das Platinenlayout wurde grundlegend geändert. Früher war über der Hauptplatine die standardmäßige Steuerplatine des 1571-Laufwerks eingebaut (Bild 2). Beim C128D-Blech hat man die gesamte Elektronik auf einer einzigen Platine zusammengefaßt.

Eine Anmerkung am Rande: Obwohl jetzt sämtliche Chips des C128 und der Floppy auf einer Platine traut vereint sind, werden die Daten zwischen ihnen immer noch auf traditionelle Weise mit dem seriellen Bus übertragen - Schade, ein paralleler Bus zum User-Port hätte viele Hardware-Erweiterungen vereinfacht.

Aber kommen wir wieder zur Sache. Es bleibt die Frage: Haben diese Änderungen für den Anwender Konsequenzen? Darauf muß man leider mit »ja« antworten, denn die Konsequenzen sind größtenteils unerfreulich.

Während man das neue Blechgehäuse und das Fehlen des Lüfters noch positiv bewerten kann und hofft, daß die zweifelsohne geringeren Fertigungskosten den Kunden zugute kommen werden, führen die Änderungen an der Hard- und Software dazu, daß der C128D-Blech nicht mehr vollständig kompatibel zu seinen Vorgängern ist.

Kompatibilitäts-Probleme

Die Inkompatibilität ergibt sich vorwiegend für die integrierte Floppystation. Es laufen alle Programme nicht mehr, die entweder sehr dicht an der Hardware programmiert sind (wie Kopierprogramme), oder die »wilde« Einsprünge ins Betriebssystem vornehmen (ohne Benutzung der Sprungtabellen).

Auch mit Hardware-Erweiterungen, die auf eine bestimmte Anordnung der Chips angewiesen sind, gibt es Probleme, weil sie rein physisch nicht mehr in das Gerät passen. So arbeitet auch außer Dolphin-DOS (laut Angabe des Herstellers) kein anderer professioneller Floppy-Speeder mit diesem Laufwerk zusammen. Alle übrigen Hersteller arbeiten jedoch an den entsprechenden Versionen, mit deren Erscheinen sicher bald gerechnet werden kann.

Ferner wird auch keine der üblichen Betriebssystem-Umschaltplatinen in diesem Computer funktionieren, da Basic- und Betriebssysteme des C 64- und C 128-Modus in einem einzigen 32-KByte-ROM untergebracht sind. Die Hersteller von Hardware-Erweiterungen werden sich sicher darauf einstellen, und es ist nur eine Frage der Zeit, wann entsprechende Umschaltplatinen auf den Markt kommen. Voraussichtlich auf der Basis des EPROMs 27512, das längst nicht jeder EPROM-Brenner programmieren kann. Doch wie sieht es mit dem DOS aus?

Betriebssystem geändert

Hauptursache für die Software-Inkompatibilität ist das geänderte Betriebssystem der eingebauten 1571. Während eine gewöhnliche 1571 mit dem DOS 3.0 ausgestattet ist, meldet sich die eingebaute Floppy mit »73,CBM DOS V3.1 1571,00,00«. Sie können diese Meldung abfragen, indem Sie im C128-Modus mit <F8> in den Monitor gehen und dann »@,UJ« eingeben.

Als Neuerung bietet dieses DOS die Möglichkeit, mit »U0>V1« das Verify nach dem Schreibvorgang abzuschalten. Dadurch kann vor allem das Speichern beschleunigt werden, allerdings auf Kosten der Datensicherheit. Mit »U0>V0« wird ein Verify wieder eingeschaltet. Außerdem wurden im neuen DOS 3.1 einige Fehler verbessert, auf die Karsten Schramm in seinem Floppybuch bereits hingewiesen hat.

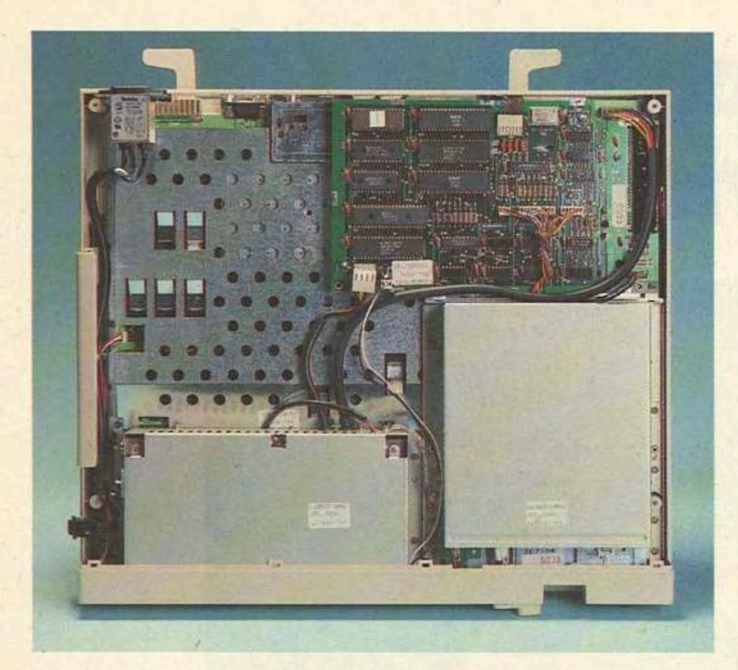


Bild 2. Beim alten C128D ist die übliche Steuerplatine der Floppy 1571 über der Hauptplatine eingebaut.

Durch diese Korrekturen und einige Erweiterungen wurden vor allem die MFM-Routinen (aber auch andere) um einige Byte versetzt. Als Folge springt beispielsweise ein Programm, das die bisher bei $877C gelegene Routine zum Einschalten der LED benutzt, jetzt mitten in eine andere Routine und stürzt gnadenlos ab.

Aha, es liegt also alles an dem neuen DOS. Kann man da nicht einfach das alte DOS einsetzen? Nein, leider nicht, denn Commodore hat nicht nur die Software (das Betriebssystem), sondern auch die Hardware bei der Floppy entscheidend verändert (wenn schon, denn schon). Das Schieberegister des einen Ein-/Ausgabe-Bausteins CIA 6526 benutzte man bisher in der 1571 lediglich für den schnellen seriellen Bus. Eigentlich ist dieser Baustein für eine so geringe Ausnutzung viel zu teuer. Auch Commodore kam zu dieser Erkenntnis, und er wurde durch einen sogenannten Custom-Chip ersetzt.

Neuer I/O-Chip

Dieser Custom-Chip, der gleichzeitig weitere ICs ersetzt, emuliert recht gut das Schieberegister bei $400C. Auch die Adressen $400D und $400E haben ihre Funktion beibehalten. Jedoch die Initialisierung dieses Chips ist völlig anders, da er weder einen Timer, noch eine Echtzeituhr oder Parallelports besitzt. Das alte DOS 3.0 kann daher nicht mit diesem Chip zusammen verwendet werden.

Welche Schlußfolgerungen lassen sich nun aus all dem ziehen?

1) Benutzen Sie beim Programmieren wirklich nur die offiziellen Commodore-Sprungtabellen oder schreiben Sie, falls dies nicht möglich ist, lieber eine kurze Routine selbst, statt wild im Betriebssystem herumzuspringen.

2) Wenn Sie ein Programm oder beispielsweise einen Floppyspeeder kaufen, erkundigen Sie sich nachdrücklich, ob das Produkt mit dem neuen C 128D-Blech (mit DOS 3.1) kompatibel ist, und ob man es andernfalls wieder zurückgeben kann.

(Günther Jilg/kn)

Welcher Floppy-Speeder ist der richtige?



64ER ONLINE

Auf dem Markt sind zur Zeit so viele verschiedene Floppy-Speeder erhältlich, daß sogar der aufmerksame Kunde leicht den Überblick verliert. Wir zeigen Ihnen aus diesem Grund, was Sie beim Kauf eines Floppybeschleunigers alles beachten müssen und geben Ihnen eine Übersicht über die gängigsten Systeme.

Prinzipiell ist Einkaufen recht einfach. Man muß nur wissen, was man sich eigentlich zulegen will. Und gerade das ist das Problem bei Floppy-Speedern. Der eine Speeder lädt schneller als die anderen; dafür funktionieren mit ihm weniger Programme. Das eine Produkt erfordert einen kompletten Umbau von Computer und Diskettenlaufwerk; ein anderes funktioniert durch einfaches Einstecken in den Expansion-Port des C 64 oder C 128.



Bild 1. Die drei Module für den Expansion-Port: Magic-Formel, Dela-DOS und Hypra-Disk.

Ein Beschleunigungssystem läuft nur mit der »alten« 1541, während das Konkurrenzprodukt auch in die neue 1541c eingebaut werden kann.

Man könnte die Liste dieser kaufentscheidenden Argumente noch eine ganze Weile weiterführen. Wir wollen Ihnen die Qual der Wahl jedoch erleichtern und nicht erschweren. Aus diesem Grund bekommen Sie im folgenden ein paar Tips, die Ihnen eine Menge unangenehmer Überraschungen ersparen können.

Bevor Sie sich für ein spezielles Beschleunigungssystem entscheiden, ist es natürlich von grundlegender Bedeutung, welche Computeranlage Sie zu Hause stehen haben. Handelt es sich um einen C 64 mit einer 1541, oder besitzen Sie einen C 128D mit zwei zusätzlich angeschlossenen 1571-Laufwerken? Ist eventuell ein Drucker vorhanden? Verwenden Sie eine 80-Zeichen-Karte, ein EPROM-Programmiergerät oder ein Hardcopy-Modul?

Alle diese Fragen müssen Sie sich vor dem Kauf eines Floppy-Speeders stellen, da es verschiedene Systeme gibt, die unter Umständen mit der einen oder anderen Erweiterung nicht zusammenarbeiten. Wenn Sie Ihren C 64 erst in den letzten Monaten neu gekauft haben, ist Ihr Computer wahrscheinlich mit der neuen C 64-Platine ausgerüstet. Hier sind Kernel und Basic-Interpreter in einem Baustein untergebracht (früher in zwei). Dazu ein Tip: Geben Sie bei der Bestellung eines Speeders das Kaufdatum des Computers mit an und bei welchem Händler Sie ihn gekauft haben. Auch beim C 128D hat sich in jüngster Zeit etwas geändert (siehe Seite 24).

Besitzen Sie zum Beispiel eine 80-Zeichen-Karte für die Bildschirmausgabe, die im Expansion-Port steckt, dann können Sie mit fast 100prozentiger Sicherheit davon ausgehen, daß ein am Expansion-Port betriebener Floppy-Speeder nicht läuft. Verwenden Sie einen EPROMer am

User-Port, dann kommt sicherlich kein Floppy-Beschleuniger in Betracht, dessen Parallelkabel ebenfalls am User-Port angeschlossen wird.

Damit Sie genau wissen, welcher Speeder für Sie in Frage kommt, haben wir in den Tabellen mit den technischen Daten jeweils auch die Hardware-Konfiguration des betreffenden Systems abgedruckt. Generell gilt: In den seltensten Fällen vertragen sich Hardware-Erweiterungen für den Computer, wenn sie alle am gleichen Anschluß (User- oder Expansion-Port) eingesteckt werden müssen. Wenn Sie sich also nicht ganz sicher sind, ob Ihre bisherigen Erweiterungsplatinen auch weiterhin einwandfrei arbeiten, sollten Sie besser vorher beim Händler fragen, ob sich der Speeder mit Ihrer Gerätekonfiguration verträgt.

Die Sache mit der Software

Neben der entsprechenden Hardware zur »Wertsteigerung« Ihres Computers besitzen Sie natürlich auch noch Software, die Sie häufig benutzen. Und hier kommt auch schon das zweite Problem beim Kauf eines Floppy-Speeders auf Sie zu: Läuft er mit den vorhandenen Programmen oder »stürzt« das System ab? Natürlich ist es uns im folgenden Test nicht möglich, alle Beschleuniger mit allen Programmen auf Lauffähigkeit durchzutesten. Wir haben uns aber ein paar weit verbreitete und kritische Programme herausgesucht, die es zumindest im kleinen Rahmen gestatten, eine Aussage über die Verträglichkeit eines Speeders zu machen. Verwenden Sie ein bestimmtes Programm ziemlich oft, das jedoch nicht in unserer Liste auftaucht, dann sollten Sie sich mit dem Händler des gewünschten Floppy-Speeders in Verbindung setzen.



Bild 2. Speeddos: das verbreitetste System für die 1541

Neben den bisher genannten Problemen mit den Beschleunigungssystemen kommen noch ein paar weitere Entscheidungskriterien hinzu, die Sie sorgfältig durchdenken sollten. Da ist einmal die Geschwindigkeit. Es gibt sehr aufwendige Systeme, die sämtliche Diskettenoperationen beschleunigen. Andere beschränken sich auf schnelleres Laden und Speichern von Programmen. Was von Ihnen im einzelnen benötigt wird, müssen natürlich Sie entscheiden. Letztendlich handelt es sich hierbei auch um eine Preisfrage. Zusätzlich sollten Sie auch die Zukunft nicht außer Betracht lassen. Planen Sie sich in nächster Zeit ein zweites Diskettenlaufwerk anzuschaffen? Dann sollten Sie aufpassen, daß sich das Beschleunigungssystem auch auf neue Zusatzgeräte erweitern läßt.

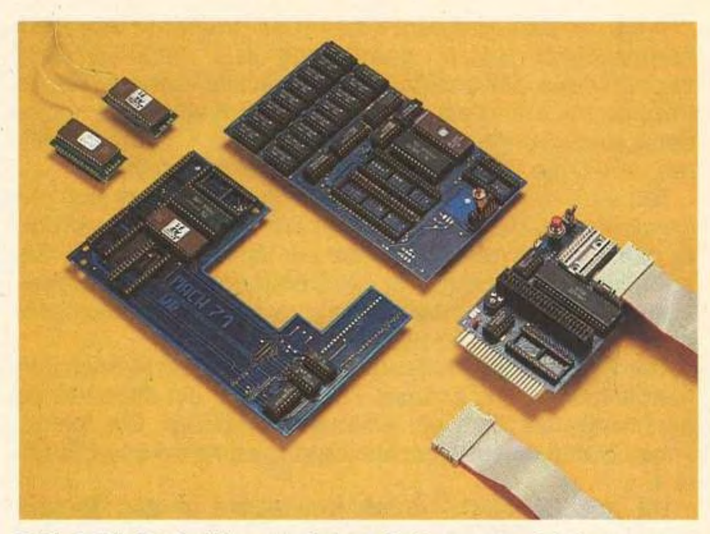


Bild 3. Die Roßmöller-Produkte: Turboaccess, Turbotrans und Mach 71 (der genaue Test von Mach 71 erfolgt im nächsten Artikel)

Bevor wir nun mit unserem Test beginnen, ein paar Worte zu den Testbedingungen. Geprüft werden die wichtigsten Diskettenoperationen, um zu einem ausgeglichenen Gesamtbild, die Geschwindigkeit betreffend, zu kommen. Wir geben keine Beschleunigungsfaktoren an, sondern teilen Ihnen lediglich die Dauer der einzelnen Vorgänge mit. Das macht die Werte übersichtlicher und beschränkt den Test auf wichtige Details.

Die Testbedingungen

Die einzelnen Zeitangaben wurden durch praxisnahe Tests ermittelt. Deshalb ist sowohl die Suche im Directory als auch die Positionierungszeit des Schreib-Lese-Kopfes der Floppystation in der gemessenen Zeit enthalten. Das gespeicherte, geladene und gelöschte 202-Block-Programm steht jeweils an erster Stelle im Directory auf einer ansonsten leeren Diskette. Die Utility-Diskette für den Validate-Test ist bis auf 40 freie Blöcke mit Hilfsprogrammen gefüllt. Die relative Testdatei besteht aus 800 Datensätzen zu je 133 Byte und wird lediglich angelegt. Ein Auffüllen mit Daten erfolgt nicht.

In der zweiten Testphase wurden zehn »harte Brokken« an Programmen ausgesucht, die unter Computerbesitzern relativ weit verbreitet sind. Diese Testprogramme zeichnen sich alle durch einen mehr oder minder hartnäckigen Kopierschutz und Autostart aus, die so manchem Floppy-Speeder das Leben schwermachen, wie wir noch sehen werden.

Im letzten Abschnitt jeder Tabelle (Tabelle 1, ganz rechts) sehen Sie eine Aufstellung der wichtigsten Eigenschaften jedes Beschleunigungssystems. Diese Liste gibt Ihnen einen groben Überblick über den Leistungsumfang jeder Erweiterung.

Weiterhin zeigen wir Ihnen auch wesentliche »Sonderausstattungen« der Beschleuniger. Diese eingebauten Zusätze gehen von einer einfachen Funktionstasten-Belegung bis hin zu eingebauten Centronics-Interfaces und Maschinensprache-Monitoren und beeinflussen ganz wesentlich den Komfort bei der Arbeit mit dem Computer.

Alle Angaben, die im Laufe des Berichts gemacht werden, beschränken sich natürlich auf die jeweils aktuelle Version eines Floppy-Speeders. Eventuelle Verbesserungen, die nach Redaktionsschluß erfolgen, können nicht berücksichtigt werden.

Wenn wir die einzelnen Systeme mit Programmen auf die Kompatibilität testen, dann heißt das natürlich nicht zwangsläufig, daß unsere Angaben für alle Versionen eines Programms zutreffen. Manchmal sind mehrere verschiedene Ausführungen auf dem Markt, die sich auch durch ihren jeweiligen Kopierschutz unterscheiden können.

Bei unseren ersten »Delinquenten« handelt es sich um Steckmodule, die Ihnen durch einfaches Einstecken in den Expansion-Port des Computers zur Verfügung stehen. Sie erfordern keinen Eingriff in den Computer oder die Floppystation, so daß die Inbetriebnahme auch vom Laien problemlos zu realisieren ist.

Das **Dela-DOS** (Bild 1) erhalten Sie in einem kleinen und handlichen Modulgehäuse. Zusätzlich ist ein doppelseitig beschriebenes Blatt als Anleitung beigefügt. Die technischen Daten dieses Beschleunigungssystems zeigt Tabelle 1.

Da das Dela-DOS durch Einstecken in den Expansion-Port des Computers aktiviert wird, verwendet es den seriellen Bus für die Datenübertragung zwischen Computer und Diskettenlaufwerk. Es enthält das DOS 5.1 zum bequemen Senden von Kommandos an die Floppystation, belegte Funktionstasten für komfortables Arbeiten, erweiterte Tastenfunktionen zur Cursorsteuerung, eine Centronics-Schnittstelle am User-Port des Computers und eine Hardcopy-Routine, die jedoch nur die Centronics-Schnittstelle anspricht. Der Preis beträgt 79 Mark, und das Modul kann am C 64, SX 64, C 128 und C 128D betrieben werden. Bei den beiden letzteren Computern ist die Erweiterung jeweils im C 64-Modus aktiv.

Das **Hypra-Disk-Modul** von Rex-Datentechnik (Bild 1) ist der preiswerteste Vertreter in unserem Vergleichstest. Es kostet nur 29,95 Mark und beschleunigt lediglich das Laden von Programmen. In Tabelle 1 sehen Sie die Leistungsdaten dieses Minimalsystems zusammengefaßt. Für 39,95 Mark ist eine Version von Hypra-Disk-Modul mit Funktionstasten-Belegung erhältlich.

Bei unserem nächsten Kandidaten kann von einem Minimalsystem hingegen keine Rede sein. Es handelt sich um **Magic-Formel V2.0** (Bild 1) von Grewe Computertechnik. Bei diesem Modul handelt es sich nicht nur um ein Beschleunigungssystem, sondern vielmehr um ein Riesen-Toolkit für den aktiven Computeranwender und Programmierer. Magic-Formel enthält eine große Anzahl an Basic-Erweiterungen, einen Maschinensprachemonitor, ein eingebautes Malprogramm, einen Freezer, einen

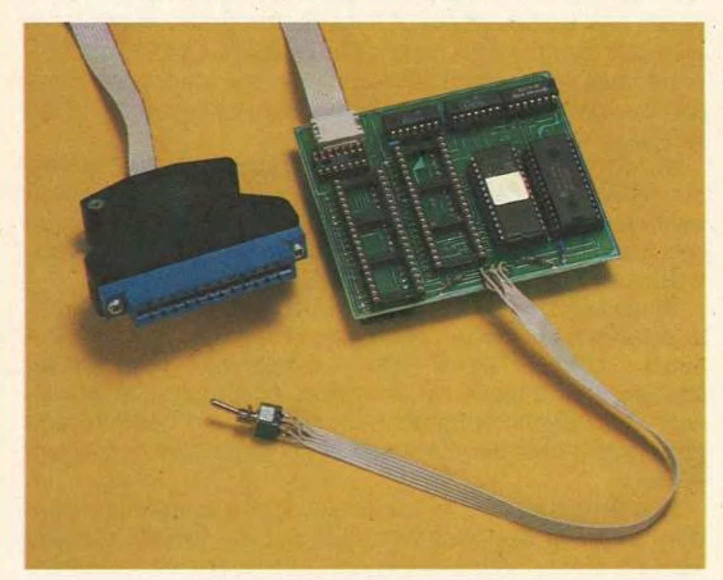


Bild 4. Dolphin-DOS: ein komfortabler Speeder mit vielen Extras

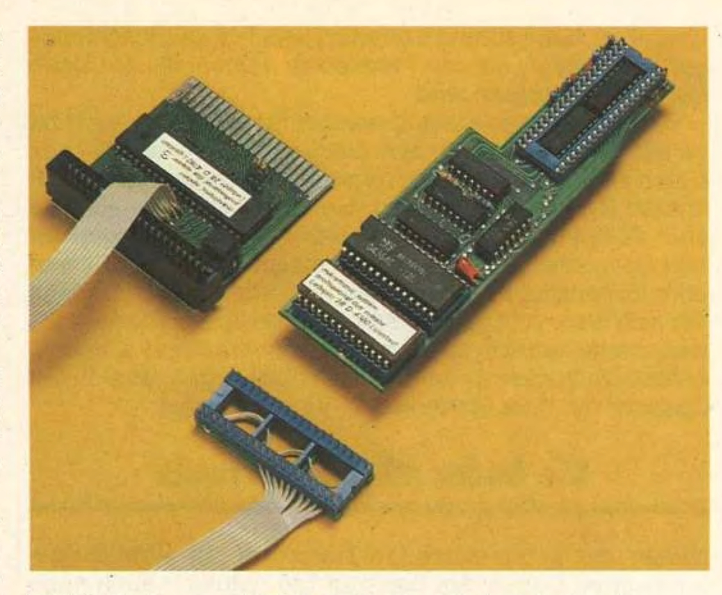


Bild 5. Professional-DOS: Hohe Geschwindigkeit und Betriebssicherheit zeichnen dieses System aus

2-Pass-Assembler, eine Pull-Down-Menü-Steuerung und eine komplette Textverarbeitung (Magic Text) sowie ein softwaregenerierter 80-Zeichen-Modus. Man muß aus diesem Grund ein wenig anders beurteilen, wenn man die Beschleunigungswerte (Tabelle 1) des eingebauten Floppy-Speeders mit den anderen Systemen vergleicht. Magic-Formel braucht sich jedoch keineswegs zu verstecken. Für einen seriellen Beschleuniger leistet das Modul eine ganze Menge. Besitzen Sie einen C 64, so sind Sie mit 169 Mark dabei. Die C 128-Version war zum Redaktionsschluß noch in der Entwicklungsphase.

Einbausystem oder Steckmodul

Bisher haben wir Ihnen drei Module für den Expansion-Port vorgestellt, die durch einfaches Einstecken betriebsbereit sind. Jetzt wollen wir zu den Einbausystemen kommen. Es darf an dieser Stelle nicht verschwiegen werden, daß diese Beschleuniger zwar schwerer als Module zu installieren sind, jedoch in der Praxis eine sehr viel höhere Software-Kompatibilität erreichen können. Das heißt es arbeiten mehr Programme reibungslos mit ihnen zusammen.

Speeddos Plus (Bild 2) ist einer der ältesten Beschleuniger für den C 64. Er arbeitet im Gegensatz zu den Steckmodulen mit einem Parallelkabel zwischen Computer und Floppystation. Das erlaubt eine höhere Geschwindigkeit beim Datenaustausch, erfordert aber dafür ein Öffnen des Diskettenlaufwerks. Auch der Computer muß zum Einbau von Speeddos Plus aufgeschraubt werden. Den Lohn für Ihre Mühe erhalten Sie mit den Leistungsmerkmalen des Floppy-Speeders (Tabelle 1). Speeddos Plus enthält mehrere Zusatzfunktionen im Betriebssystem, und sogar ein Kopierprogramm ist im Lieferumfang enthalten. Disketten lassen sich nicht nur auf 35 sondern sogar auf 40 Spuren formatieren, was mehr Speicherplatz zur Folge hat (749 statt 664 freie Blöcke). Für 149 Mark bekommt der Anwender zwei Platinen, ein Parallelkabel für den Anschluß am User-Port, eine Diskette mit dem Kopierprogramm »FCopy III« und eine 12seitige Bedienungsanleitung.

Dem eben beschriebenen System sehr ähnlich ist **Rex-DOS** (Tabelle 1). Es ist im Diskettenlaufwerk fast vollständig kompatibel zu Speeddos Plus; lediglich im Computer fehlen einige Zusatzfunktionen des erweiterten Speeddos-Betriebssystems. Rex-DOS ist für 99,50 Mark zu haben und

enthält in seinem Lieferumfang zwei Platinen, ein Parallelkabel und eine mehrseitige Anleitung.

Bild 3 zeigt mehrere Beschleunigungssysteme von Roßmöller. Eines davon ist **Turboaccess**, das ebenfalls schon sehr lange auf dem Markt ist. Tabelle 1 zeigt die technischen Daten dieses Speeders. Früher wurde das Parallelkabel im Gegensatz zu Speeddos über eine Zusatzplatine am Expansion-Port angeschlossen (Bild 3). Die neueste Version wird am User-Port des C 64 angeschlossen und kostet nur noch 99 Mark. Auch an den Anschluß für ein zweites Diskettenlaufwerk haben die Entwickler gedacht.

Mit noch besseren Leistungsdaten als die bisher besprochenen Systeme wartet **Dolphin-DOS** (Bild 4) auf. Wenn Sie sich die Zeiten in Tabelle 1 ansehen, erkennen Sie einen deutlichen Abstand zu den Daten der anderen Systeme. Das hängt damit zusammen, daß die im folgenden besprochenen Speeder einer neueren Generation angehören, die hard- und softwaremäßig weiter entwickelt sind, was sich auch am Preis bemerkbar macht, wie wir noch sehen werden.

Dolphin-DOS besitzt neben den Beschleunigungsfaktoren noch eine ganze Menge an zusätzlichen Extras im Betriebssystem des Computers. Darunter finden wir einen Maschinensprachemonitor, frei programmierbare, vorbelegte Funktionstasten, eine Centronics-Schnittstelle, eine Hardcopy-Routine, eine OLD-Funktion zum Zurückholen gelöschter Basic-Programme und die Möglichkeit, Disketten 40spurig zu formatieren und zu beschreiben. Das System kostet 178 Mark und ist teilweise Speeddos-kompatibel. Das System wird mit zwei Kopierprogrammen auf Diskette und einer Anleitung ausgeliefert.

Das **Professional-DOS** von VTS Data (Bild 5) wird im Gegensatz zu Dolphin-DOS nicht am User-Port, sondern am Expansion-Port des Computers angeschlossen. Das verhindert Konflikte zwischen dem Parallelkabel und der ebenfalls am User-Port herausgeführten Centronics-Schnittstelle. Will beispielsweise der Speeddos-Anwender zusätzlich einen Drucker mit Centronics-Interface an seinen Computer anschließen, bleibt ihm nur der Kauf einer sogenannten »User-Port-Weiche«, die zwei Anschlüsse zur Verfügung stellt. Professional-DOS besteht aus drei Platinen, wobei eine in der Floppystation, die zweite am Expansion-Port und die dritte im Inneren des Computer-Gehäuses Platz findet. Die Leistungsdaten (Tabelle 1) können sich sehen lassen.

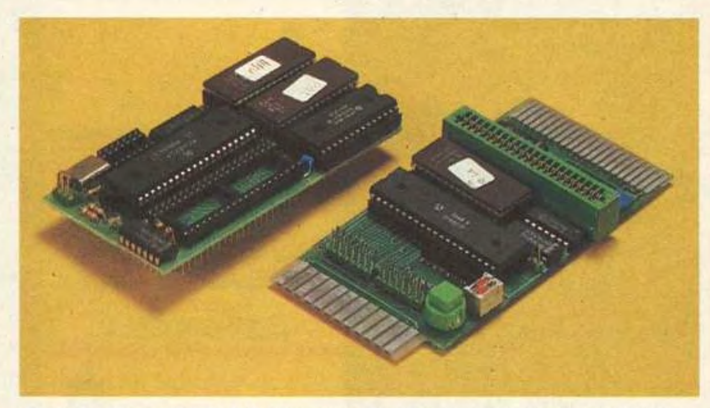


Bild 6. Prologic-DOS Classic: Ein Beschleuniger der Extraklasse mit viel Komfort

	Zeitmessung (in Sekunden)							Kompatibilitätstest (Die mit »*« gekennzeichneten Programme funktionieren)										Besonderheiten des Beschleunigers
a) Name b) Firma c) Preis	LOAD (202 Blöcke)	SAVE (202 Blöcke)	SCRATCH (202 Blöcke)	VALIDATE (Utility-Disk)	FORMATIEREN (35 Spuren)	relative Datei erzeugen (423 Blöcke)	relative Datei löschen (423 Blöcke)	Ali	Alternate Reality	Arkanoid	Bulldog	Font Master II	Hypaball	Profi-Painter	Psi-5	Textomat Plus	The Sentil	
a) Dela-DOS b) Dela Elektronik, Merkenicher Straße 87-89, 5000 Köln 60, Tel. 0221/71517-20 bis 22 c) C 64, SX64, C 128, C 128D (im C 64-Modus): 79 Mark	25,4	25,2	23,7	113,4	81,5	222,6	53,5	—	*	—	—	—	—	*	—	*	*	belegte Funktionstasten, DOS 5.1, erweiterte Cursorfunktionen, eingebaute Centronics-Schnittstelle, Hardcopy-Routine, kein Eingriff in Computer oder Floppystation
a) Hypra-Disk-Modul b) Rex-Datentechnik, Stresemannstr. 11, 5800 Hagen 1, Tel. 02331/32734 + 16979 c) C 64, SX64, C 128, C 128D (im C 64-Modus): 29,95/39,95 Mark	23,9	139,0	25,2	114,0	85,8	222,0	52,8	—	—	*	*	—	—	*	—	—	—	kein Eingriff in Computer oder Diskettenlaufwerk, für 39,95 Mark mit Funktionstasten-Belegung
a) Magic Formel V2.0 b) Michael Grewe Computertechnik GmbH, Richard-Wagner-Str. 73, 4350 Recklinghausen, Tel. 02361/181354 c) C 64, SX64: 169 Mark	8,2	9,7	9,9	75,3	32,7	22,6	16,9	*	—	—	—	—	—	*	—	*	—	kein Eingriff in Computer oder Diskettenlaufwerk, 64 KByte ROM, eigene 8 KByte RAM, Basic-Erweiterungen, Freezer, Malprogramm, Maschinensprache-Monitor, Assembler, Centronics-Schnittstelle, Hardcopy-Routinen, Pull-down-Menü-Steuerung, 80-Zeichen-Modus, Textverarbeitung

Tabelle 1. Die technischen Daten sämtlicher Speeder auf einen Blick. Bitte beachten Sie, daß der Kompatibilitätstest bei Steckmodulen zwangsläufig schlechter ausfällt, da die Gerätekonfiguration zusätzlich Probleme aufwirft.



a) Name b) Firma c) Preis	Zeitmessung (in Sekunden) LOAD (202 Blöcke)	SAVE (202 Blöcke)	SCRATCH (202 Blöcke)	VALIDATE (Utility-Disk)	FORMATIEREN (35 Spuren)	relative Datei erzeugen (423 Blöcke)	relative Datei löschen (423 Blöcke)	Kompatibilitätstest (Die mit »*« gekennzeichneten Programme funktionieren) All	Alternate Reality	Arkanoid	Bulldog	Font Master II	Hypaball	Profi-Painter	Psi-5	Textomat Plus	The Sentil	Besonderheiten des Beschleunigers
a) Speeddos-Plus b) Electronic-Service Christoph Dichte, Fährstr. 33, 2212 Brunsbüttel c) C 64 + 1541/1541c: 149 Mark	15,0	102,9	25,9	97,8	24,0	215,0	53,4	*	—	*	*	—	*	*	—	*	*	belegte Funktionstasten, Maschinensprachemonitor, Centronics-Schnittstelle, DOS 5.1, erweiterte LIST-Funktion, verschiedene Zahlensysteme, Hardcopy-Funktion, 35/40 Spuren, Kopierprogramm FCOPY III im Lieferumfang enthalten
a) Rex-DOS b) Rex-Datentechnik, Stresemannstr. 11, 5800 Hagen 1, Tel. 02331/32734 + 16979 c) C 64 + 1541/1541c: 99,50 Mark	15,4	103,1	25,8	97,6	24,2	213,7	53,4	*	—	*	*	—	*	*	—	*	*	belegte Funktionstasten, DOS 5.1, 35/40 Spuren
a) Turboaccess 2.7+ b) Roßmöller CT, Neuer Markt 21, 5309 Meckenheim, Tel. 02225/2061 c) C 64 + 1541: 99 Mark	21,9	101,4	23,1	97,6	18,0	209,3	47,6	*	*	*	—	—	*	*	—	*	*	Centronics-Schnittstelle, Anschluß für zweites Laufwerk eingebaut
a) Dolphin-Dos 2.0 b) Jan Bubela, Egenolffstr. 19, 6000 Frankfurt/Main, Tel. 069/446573 c) C 64 + 1541/1541c, SX64, C 128: 178 Mark	5,2	9,5	4,3	16,2	19,0	66,6	7,3	—	—	*	—	—	—	*	*	*	—	teilweise Speeddos-kompatibel, belegte Funktionstasten, Maschinensprache-Monitor, Arbeiten mit verschiedenen Zahlensystemen, stark erweiterte Reset-Routine im Computer, Speeder stufenweise abschaltbar, Centronics-Schnittstelle, 35/40 Spuren
a) Professional-Dos für C 64 und 1541 b) VTS Data GmbH, Hauptstr. 48, 5014 Kerpen 1, Tel. 02234/71601 c) C 64 + 1541/1541c User-Port: 197 Mark, Expansion-Port: 258 Mark	3,6	8,9	6,3	16,2	18,2	39,6	13,0	*	—	*	*	—	—	*	—	*	*	belegte Funktionstasten, Centronics-Schnittstelle, mehrere Zahlensysteme, variable Taktfrequenz im Diskettenlaufwerk, Hardcopy-Routine, User-Port bleibt frei
a) Prologic-Dos Classic Rex b) Rex-Datentechnik, Stresemannstr. 11, 5800 Hagen 1, Tel. 02331/32734 + 16979 c) C 64 + 1541/1541c: 198 Mark Variante am User-Port: 149,50 Mark	4,5	10,4	4,6	14,7	20,2	44,5	9,8	—	—	—	—	*	*	—	*	*	*	belegte Funktionstasten, User-Port bleibt frei, Centronics-Schnittstelle mit eigenem Anschluß, Anschluß für zweites Laufwerk eingebaut, Kopierprogramme in ROM-Disk, stufenweise abschaltbares Beschleunigungssystem, Dos 5.1
a) Turbotrans 3.4 b) Roßmöller CT, Neuer Markt 21, 5309 Meckenheim, Tel. 02225/2061 c) C 64 + 1541/1541c: 298 Mark C 128 + 1541/1541c: 349 Mark	8,9/ 2,4*	100,1/ 20,9*	22,7/ 2,6*	94,0/ 7,9*	18,3/ <1*	206,5 6,5*	47,3/ 5,1*	*	*	*	*	—	*	*	—	—	*	eingebaute RAM-Floppy, User-Port bleibt frei, Centronics-Schnittstelle, Anschluß für zweites Laufwerk eingebaut, Maschinensprachemonitor, erweiterte Reset-Routine des Computers, Hardcopy-Routine

sen und erlauben ein angenehmes und schnelles Arbeiten mit dem Diskettenlaufwerk. Neben der Hardware bekommt der Anwender für die 258 Mark noch eine mehrseitige Anleitung und eine Diskette mit speziell auf das System zugeschnittenen Kopierprogrammen und Utilities. Für 197 Mark ist auch eine User-Port-Version erhältlich.

Ebenfalls schnell und komfortabel läßt sich mit **Prologic-DOS Classic** (Bild 6) arbeiten. Dieses System wird seit neuestem nur noch von Rex-Datentechnik vertrieben und hat serienmäßig eine ROM-Disk mit nützlichen Programmen eingebaut. Die ROM-Disk befindet sich im Diskettenlaufwerk, und von ihr können zum Beispiel die mitgelieferten Kopierprogramme mit LOAD geladen werden, ohne daß eine Diskette eingelegt sein muß. Wie schon bei Professional-DOS, wird auch bei Prologic-DOS Classic das Parallelkabel mit einer Adapterplatine an den Expansion-Port des Computers angeschlossen. Auf dieser Platine befindet sich zusätzlich ein eigener Anschluß für einen Centronics-Drucker, so daß der User-Port bei Prologic-DOS Classic vollkommen freibleibt und beispielsweise für DFÜ verwendet werden kann.

Wie bei Turboaccess ist auch bei Prologic-DOS Classic der Anschluß für ein zweites Diskettenlaufwerk vorhanden, so daß lediglich die Platine für die zusätzliche Floppystation nachgekauft werden muß, wenn mit zwei Laufwerken gearbeitet werden soll. Das System kostet 198 Mark. Man bekommt dafür drei aufwendige Platinen, das Parallelkabel und ausführliche Unterlagen. Seit neuestem gibt es auch eine User-Port-Version (Prologic-DOS U) für 149,50 Mark.

Turbotrans (die Daten zeigt Tabelle 1) ist der bei weitem aufwendigste Speeder, den wir Ihnen in diesem Vergleichstest vorstellen wollen. Dieses System rüstet den Speicher im Diskettenlaufwerk auf 256 oder 512 KByte RAM auf und erlaubt somit das Einlesen eines oder zweier kompletter Disketteninhalte. Im Speicher können die Disketteninhalte dann schnell und ohne mechanische Beanspruchung des Laufwerks bearbeitet und schließlich wieder auf die Diskette(n) zurückgeschrieben werden. Da sich die neueste Hardware-Version gegenüber der in Bild 3 abgebildeten verändert hat, kann das Floppy-Gehäuse nach dem Einbau auch geschlossen werden (mit kleinen Abstandsröllchen). Es kostet zusammen mit dem 50seitigen Handbuch und einer Diskette mit Kopierprogrammen 298 Mark. Besitzen Sie einen C 128 mit der Floppy 1541, so erhöht sich der Preis auf 349 Mark.

Wenn Sie nun den Wunsch hegen, sich ein Beschleunigungssystem zuzulegen, dann können Sie durch Vergleiche zwischen den zwölf Tabellen mit den technischen Daten der Speeder, das für Sie richtige System finden. Die kaufentscheidenden Kriterien sind in jeder Tabelle in Stichpunkten aufgeführt und lassen Sie einen Überblick über das Leistungsangebot gewinnen. Sind noch Zweifel bezüglich Kompatibilitätsfragen oder Lieferumfang vorhanden, dann wird Ihnen bestimmt jeder Hersteller oder Händler eines Beschleunigungssystems gerne Auskunft zu Ihren Fragen geben. Abschließend wollen wir Sie nochmals darauf hinweisen, daß es sowohl den C64 als auch die Floppy 1541 mit verschiedenen Platinen-Varianten gibt. Bitte geben Sie daher beim Kauf eines Floppy-Speeders unbedingt an, wann und wo Sie die Gerätekonfiguration gekauft haben. Die Hardware-Hersteller haben sich nahezu alle auf die verschiedenen Gerätevarianten eingestellt. (ks/kn)

Im Vergleich: Neue Speeder für die 1571

Ihr C 128 lädt Ihnen zu langsam? Wenn Sie bisher neidisch auf die für den C 64-Modus erhältlichen Floppy-Speeder geblickt haben, heißt es nun aufpassen: Vier Beschleuniger für den C 128D oder C 128 mit der 1571 stellen sich vor.

Schon richtig, die 1571 ist im C 128-Modus ein schnelles Laufwerk — vier Beschleuniger-Systeme beweisen jedoch, daß es noch schneller geht: DolphinDOS, Professional DOS, Prospeed und Mach 71 (Bilder 1 bis 4) sind angetreten, ihre Stärken zu demonstrieren. Wir wollen sehen, ob es auch Schwächen gibt.

Die wesentlichen Eigenschaften der einzelnen Speeder für den C 128 haben wir in den nächsten Abschriften zusammengefaßt.

Vorab noch ein wichtiger Hinweis: Außer DolphinDOS arbeitet noch kein Speeder mit dem neuen C 128D (mit Blechgehäuse) zusammen, da in diesem Gerät erhebliche Hardware-Änderungen vorgenommen wurden (siehe auch Seite 16).

Einen noch ausführlicheren Testbericht finden Sie in der 64'er-Ausgabe 1/88. Die Tabelle 1 zeigt die Vor- und Nachteile der vier Speeder nochmals in einer Übersicht.

In der alphabetischen Reihenfolge taucht zuerst Dol-

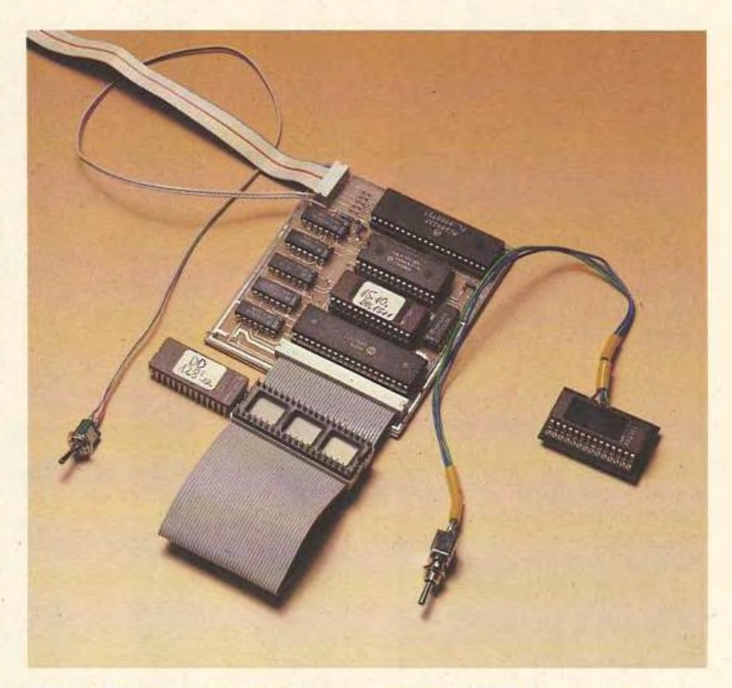


Bild 1. Schnell und auch zum neuen C 128D kompatibel: DolphinDOS 128



Bild 2. Lädt ungeschützte Programme nach der Konvertierung am schnellsten: Professional DOS

phinDOS auf (Bild 1). Der Einbau in unseren C 128D (wegen des »D« auch liebevoll »Diesel« genannt) gestaltete sich zunächst problematisch, da die Platine des Beschleunigers am Platz des Floppy-Prozessors eingesetzt wird. Leider ist dieser nicht in jedem C 128D oder in jeder 1571 gesockelt und muß dann zuerst ausgelötet werden. Ein Flachbandkabel führt von dieser Platine zum User-Port. Noch zwei EPROMs in den Computer einsetzen, eines für den C 64-, eines für den C 128-Modus, und der Einbau ist erledigt.

Der bewährte Beschleuniger überzeugt durch gute Geschwindigkeiten und viele Extras im C 64-Modus. Hier liegen drei Betriebssysteme vor, zwischen denen Sie umschalten können. Enthalten ist das originale Betriebssystem, das normale DolphinDOS-Kernel sowie eine abgespeckte Version, in die aus Gründen der Kompatibilität nur die schnellen Laderoutinen integriert wurden.

Schlichtheit contra Luxus

Um den C 64-Modus geht es hier nur sekundär, wesentlich ist, was für den C 128 getan wurde. Sehr angenehm fällt auf, daß die Formaterkennung (1541- oder 1571-Diskette) und das Suchen nach einem Boot-Sektor erheblich verkürzt ist. Ansonsten wurden keine Veränderungen am Basic- oder Betriebssystem-ROM des C 128 vorgenommen. Schade eigentlich, denn auch wenn die Kompatibilität zu bestehender Software erhöht wird, muß der stolze Besitzer auf einigen zusätzlichen Komfort verzichten.

DolphinDOS 128 kostet 198 Mark beziehungsweise 178 Mark für ein zweites Laufwerk. Für Leser, die diesen Floppy-Beschleuniger bereits für die 1541 besitzen, ist es sicher interessant zu erfahren, daß sie durchaus eine 1541 und eine 1571, die DolphinDOS enthalten, mit dem gleichen Betriebssystem betreiben können.

Kommen wir zum nächsten Kandidaten: Professional-DOS (Bild 2). Auch dieser Floppy-Beschleuniger erwartet einen gesockelten Prozessor. Das System ist teilweise überraschend schnell, sowohl im C 64- als auch im C 128-Modus. Es gibt eine Version zum Anschluß an den User-Port und eine für den Expansions-Port. Beide Versionen sind nahezu identisch, die Userport-Version erreicht allerdings durch einen neuen Datei-Typ, (»FPRG«) noch deutliche Geschwindigkeitsgewinne. Ein weiterer Vorteil dieser Ausführung ist der günstigere Preis, wohingegen Sie mit der technisch aufwendigeren (und teureren) Expansion-Port-Version gleichzeitig noch ein Modem oder einen Akustikkoppler betreiben können.

Ein neuer Datei-Typ

Weiter kommen bei beiden Ausführungen noch eine Menge Sonderfunktionen hinzu, die das Arbeiten mit dem C 128 sehr erleichtern. Wer häufiger im C 64-Modus arbeitet, wird sicherlich erfreut zur Kenntnis nehmen, daß er nach einem Reset in diesem Modus bleiben kann — ohne zusätzliches Drücken der Commodore-Taste. Erfreulich ist auch, daß der Zehnerblock rechts auf der Tastatur ständig aktiv ist. Im C 128-Modus ist das Laden aus dem Directory mit DLOAD oder BLOAD ohne lästiges Setzen des Doppelpunktes sehr angenehm. In Verbindung mit <CTRL> lassen sich vier weitere Funktionen mit den Funktionstasten ansprechen. So läßt sich der C 64-Modus über einen Tastendruck einschalten, ebenso schnell und einfach wird ein BOOT ausgeführt oder die Standardadresse des Laufwerks auf 9 geschaltet. Danach beziehen sich alle Disketten-Operationen auf eine zweite Diskettenstation.

Professional DOS kostet für den C 128 mit einer 1571 als Expansion-Port-Version 258 Mark. Die User-Port-Version kostet, trotz ihrer Geschwindigkeits-Vorteile, nur 198 Mark.

Prospeed (Bild 3), der Nachfolger von Prologic DOS für den C 128, ist vor kurzem erst im 64'er-Magazin, Ausgabe 10/87, ausführlich getestet worden. Um Ihnen aber den Vergleich mit den anderen Beschleunigern zu erleichtern, wollen wir Wesentliches noch einmal darstellen. Herausragend sind zweifellos die eingebauten Kopierprogramme: Im C 64-Modus laden Sie mit »LOAD" #B"« beispielsweise blitzschnell (es wird nicht von Diskette, sondern direkt aus dem ROM des Laufwerks gelesen) ein auf die Fähigkeiten von Prospeed abgestimmtes Backup-Programm. Ein weiteres Glanzlicht ist die aufwendige Hardware, für die zum einen ein Auslöten des Laufwerk-Prozessors nicht nö-

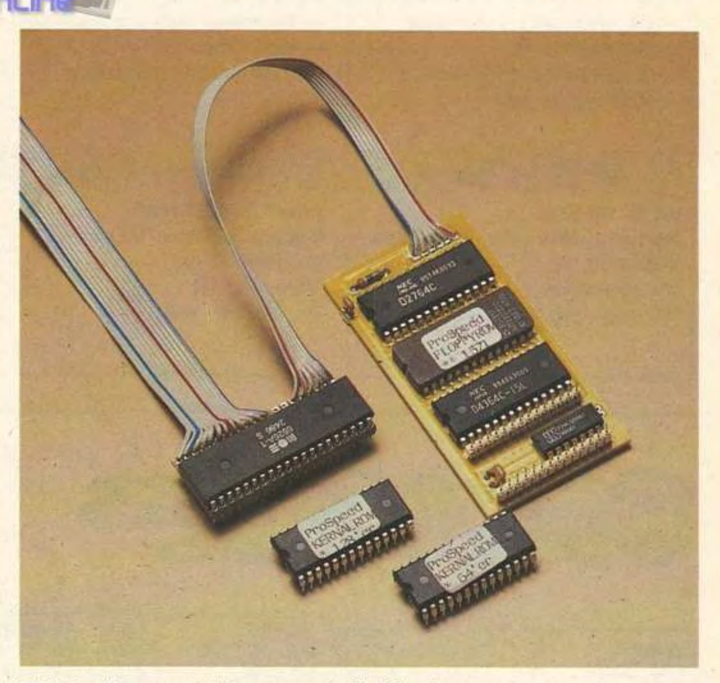


Bild 3. Eine gute Idee: spezielle Kopierprogramme im Floppy-ROM werten Prospeed deutlich auf

tig ist, und die zum anderen die 1571 deutlich kompatibler zu C 64-Programmen macht. Dies wird folgendermaßen erreicht: Die erweiterte Zeropage wird bei der Floppy 1541 ab Adresse $2000 gespiegelt, bei der 1571 jedoch nicht. Einige (unserer Meinung nach unsauber programmierte) Programme greifen auf diese Spiegel-Adressen zu. Prospeed richtet diese Spiegel-Adressen auch bei der 1571 ein.

Starker C64-Modus

Ähnlich wie Professional DOS aktiviert Prospeed den zusätzlichen Zehnerblock der C 128-Tastatur auch im C 64-Modus. Einige kleine Gags sind zusätzlich eingebaut: So ist es im C 128- genau wie im C 64-Modus erlaubt, mit »SYS 64738« einen Reset auszulösen. Prospeed kostet zur Zeit 238 Mark.

Mach 71 (Bild 4) ist als einer der ersten Floppy-Beschleuniger für den C 128 der Oldtimer unter den hier vorgestellten Systemen. Einige Neuigkeiten und die hochwertige Verarbeitung der Hardware machen dieses System auch heute noch interessant.

Die neueste Version von Mach 71, für die der Prozessor des Laufwerks nicht ausgelötet werden muß, wird an den User-Port statt wie bisher an den Expansion-Port angeschlossen. Dadurch arbeitet auch dieser Floppy-Speeder mit den Commodore-RAM-Erweiterungen 1700 und 1750 zusammen.

Durch die ausgeklügelte Verbindungstechnik — Massekabel zwischen den Datenleitungen des parallelen Busses verhindern zerstörerische Induktionen — ist ein Anschluß von bis zu vier Diskettenstationen gefahrlos möglich und Störungen von außen werden abgeschirmt. Auch die RS232 ist weiterhin betriebsfähig.

Im Lieferumfang ist unter anderem ein Programm »CP/M-Patch« enthalten, das eine kleine Anpassung am CP/M 3.0 durchführt und so Geschwindigkeitsgewinne um knapp 100 Prozent erreicht — allerdings nur im 80-Zeichen-Modus.

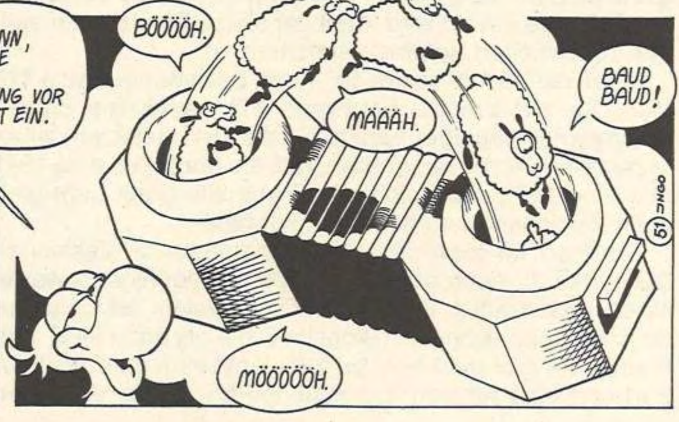
Seit Jahren schon läßt die Firma Roßmöller bei ihren Pro-

DolphinDOS 128	Professional DOS	Prospeed	Mach 71
Positiv	**Positiv**	**Positiv**	**Positiv**
— C 64-Modus mit vielen Hilfsfunktionen — drei Betriebssysteme für C 64 — günstiger Preis — schnelle Formaterkennung	— C 64-Modus Reset-fest — hohe Geschwindigkeit — Zehnerblock im C 64-Modus aktiv	— Kopierprogramme im Laufwerks-ROM — Zehnerblock im C 64-Modus aktiv — Diskettenstation wird sehr kompatibel zur 1541	— gleichzeitiger Anschluß von Beschleuniger und Akustikkoppler, Modem oder RS232-Drucker
Negativ	**Negativ**	**Negativ**	**Negativ**
— Laufwerks-Prozessor muß gesockelt sein — C 128-Modus ohne weitere Hilfsfunktionen	— Prozessor muß gesockelt sein — hoher Preis für Expansion-Port-Version	— geringe Beschleunigung bei kopiergeschützten Programmen	— interne Diskettenoperationen teilweise nicht beschleunigt
Wichtige Daten	**Wichtige Daten**	**Wichtige Daten**	**Wichtige Daten**
Preis: 198 Mark **Bezugsquelle:** Dolphin Software, Egenolfstr. 19, 6000 Frankfurt 1, Tel. 069/446573	**Preis:** 198/258 Mark **Bezugsquelle:** VTS Data, Postfach 400621, 5000 Köln 40, Tel. 02234/71601	**Preis:** 238 Mark **Bezugsquelle:** Michael Lamm, Schönbornring 14, 6078 Neu-Isenburg 2, Tel. 06102/52535	**Preis:** 259 Mark **Bezugsquelle:** Roßmöller Handshake GmbH, Maxstr. 50-52, 5300 Bonn, Tel. 0228/659980









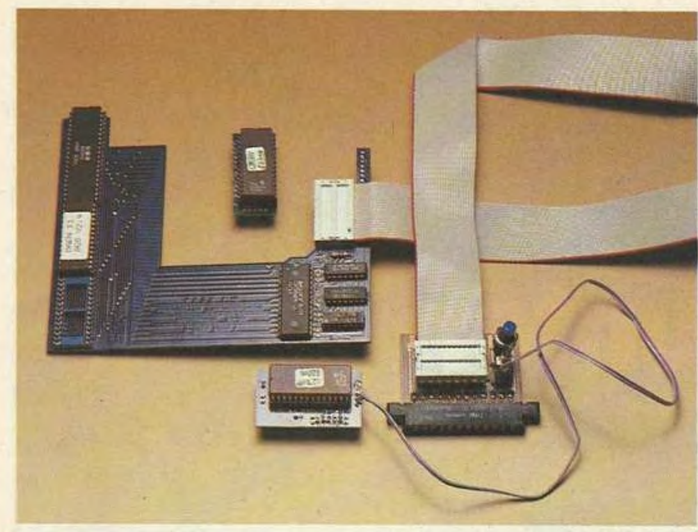


Bild 4. Erreicht mit dem professionellen Betriebssystem CP/M die besten Zeiten: Mach 71, jetzt als User-Port-Version

dukten die Funktionstasten des C 64/C 128 unberührt. Sämtliche Zusatzfunktionen werden über eine Tastenkombination mit <CTRL> angesprochen. So aktiviert zum Beispiel <CTRL C> den C 64-Modus und <CTRL D> zeigt das Inhaltsverzeichnis der eingelegten Diskette. In Planung ist ein EPROM für den freien Steckplatz des C 128, das an die speziellen Eigenschaften von Match 71 angepaßte und sehr schnelle Kopierprogramme enthalten soll.

Mach 71 gibt es in zwei Versionen, für den C 128D und für den C 128 mit einer 1571. Beide Ausführungen sind für 259 Mark erhältlich.

Welcher Laufwerksbeschleuniger ist nun der richtige? Eine Entscheidung fällt schwer. DolphinDOS ist im Durchschnitt (Laden, Speichern, Löschen, sequentielle und relative Dateien bearbeiten) das schnellste System und dazu recht preiswert. Weiter ist es der einzige Floppy-Speeder, der mit der neuesten C 128D-Verson zusammenarbeitet, er bietet aber im C 128-Modus keine Zusatzfunktionen.

Wer mit wem?

Professional DOS ist, was die reine Ladegeschwindigkeit betrifft, sicherlich extrem schnell — aber wie bei DolphinDOS muß der Laufwerks-Prozessor eventuell ausgelötet werden. Prospeed erfordert dies nicht, ist aber im Durchschnitt etwas langsamer.

Noch etwas weniger beschleunigt Mach 71, hat aber beim Betrieb mit CP/M die Nase vorn und: der Betrieb eines Akustikkopplers oder Modems ist zusammen mit dem Beschleuniger möglich. Ferner kann, da der Anschluß nun am User-Port erfolgt, wie bei den anderen Speedern mit den RAM-Erweiterungen 1700 und 1750 gerarbeitet werden.

Bedauerlicherweise ist außer DolphinDOS keiner der getesteten Beschleuniger in der Lage, mit dem neuen C 128D (im Blechgehäuse) zusammenzuarbeiten. Alle Hersteller arbeiten jedoch angeblich an entsprechenden Versionen, die in den nächsten Wochen auf den Markt kommen sollen. Die Unterschiede zwischen der 1571 und dem neuen Laufwerk sind im Artikel »Ein Problemkind: die neue Floppy im Blechgewand« auf den Seiten 16 und 17 dieser Ausgabe genauer beschrieben. Übrigens ist Mach 71 der einzige Speeder, der für den C 128D (alt) eine andere Hardware benötigt, als für einen C 128 mit einer 1571. Sie sollten beim Kauf sichergehen, daß Sie die für Ihr System richtige Ausführung erstehen. (ap/kn)



Endlich gibt es einen Floppy-Monitor, der die zahlreichen speziellen Vorteile des C 128 voll ausnützt und dabei sehr komfortabel und anwenderfreundlich zu bedienen ist.

Der größte Nachteil der vielen Floppy-Monitore für den C 64 ist wohl die begrenzte Zeichenzahl auf dem Bildschirm. Denn auf einer Matrix von 40 * 25 Zeichen läßt sich ein Sektor nicht komplett im Hex- und ASCII-Dump darstellen, und die Kompromißlösungen sind meistens unzufriedenstellend und gehen sehr auf Kosten der Benutzerfreundlichkeit. Beim C 128 ist dieses Problem jedoch nicht mehr gegeben, 80 Zeichen pro Zeile sind für diesen Zweck ideal. Bei TOP-FLOP gerät der 40-Zeichen-Bildschirm jedoch nicht ins Vergessen – er wird für einen Hilfstext benützt, der dem Anwender jederzeit durch Umschalten des Monitoradapters zur Verfügung steht.
Top-Flop ist in Basic geschrieben und wird durch mehrere Maschinenroutinen sinnvoll unterstützt. So fallen Unterprogramme, die sonst extrem zeitaufwendig sind (Floppy-Zugriffe, Darstellung des Sektors, Hardcopy etc.), kaum noch ins Gewicht. Jedoch dürfte es durch den modularen Aufbau trotzdem noch möglich sein, das Programm zu verstehen und eventuell nach eigenen Vorstellungen abzuändern.

Das Hauptprogramm (Listing 1) lädt die Maschinenroutinen (Listing 2) in den reservierten Grafikspeicher ab $2200, was mehrere Vorteile hat:

Erstens nehmen sie auf diese Weise keinen anderen, vieleicht anderweitig benötigten Speicherplatz ein (das Programm benutzt ja keine Grafik), und zweitens ist so die Kompatibilität zu allen Basic-Compilern gesichert.

Starten des Programms

Das Programm wird mit

```
RUN "TOP-FLOP.BAS" <RETURN>
```

geladen und gestartet, wobei die Taste <40/80 DISPLAY> gedrückt sein sollte. Nun meldet sich Top-Flop mit der Angabe des Laufwerkstyps (»1570« auch bei angeschlossener 1571 mit nicht zweiseitig formatierter Diskette). In der anschließenden kurzen Wartezeit wird das Programm initialisiert und das Unterprogramm »TOP-FLOP.SUB« nachgeladen. Sollte dieses auf der eingelegten Diskette nicht auffindbar sein, gibt Top-Flop eine Fehlermeldung aus.

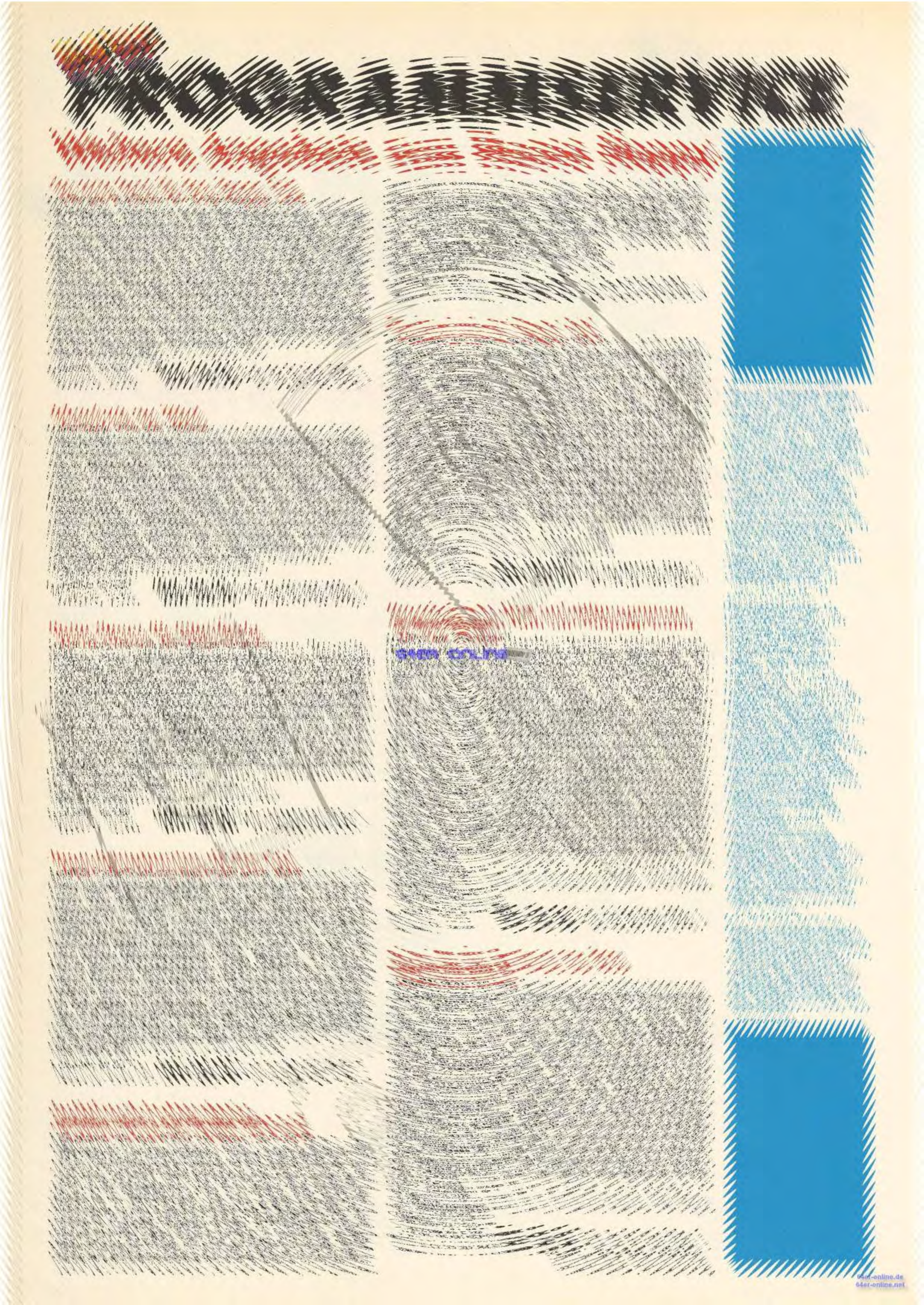
Nach dem Laden erscheint der noch leere Arbeitsbildschirm und die Diskette im Laufwerk wird initialisiert (nur bei 1570/71 – zweimaliges Rattern im Laufwerk). Jetzt ist auch parallel auf dem 40-Zeichen-Bildschirm eine Liste aller gültigen Programmbefehle in alphabetischer Reihenfolge zu sehen (Bild 1). Der Arbeitsbildschirm teilt sich folgendermaßen auf:

Übersichtliche Bildschirmaufteilung

In der ersten Zeile stehen Programmname und -art. In der nächsten stehen die Informationen über den aktuellen Track und Sektor, den zuletzt gelesenen Track und Sektor (eine Information, die Einsteiger wie Fortgeschrittene gleichermaßen zu schätzen wissen) und der gerade aktive Anzeigemodus, über den Sie noch im Rahmen des Befehls »M« mehr erfahren werden (Bild 2). In dem folgenden abgegrenzten Feld wird nach jeder Diskettenoperation der Diskstatus im üblichen Format angezeigt. Das große Hauptfeld ist für die Darstellung des Sektors beziehungsweise für das Directory und das Floppybefehle-Untermenü reserviert. Am unteren Rand des Bildschirms ist noch der Autoren- und Copyrightvermerk zu lesen, der aus Gründen der Übersichtlichkeit durch Invertierung dem Rahmen angepaßt wurde.

Nach der Initialisierung des Programms ertönt ein »Gong«, Sie befinden sich im Befehlsmodus. Sie können nun einen beliebigen Top-Flop-Befehl eingeben. Diese Befehle bestehen aus einzelnen prägnanten Buchstaben, die das Merken des Kommandos erheblich vereinfachen. Sie


64ER ONLINE



können bei der Benutzung des Programms praktisch kaum etwas falsch machen, da unkorrekte Eingaben mit einem kurzen Brummton quittiert werden. Richtige Eingaben und Beendigungen von Routinen bestätigt Top-Flop mit dem bekannten Klingelzeichen CHR$(7). Es ist also zu empfehlen, bei der Arbeit mit dem Programm immer den Lautsprecher angeschaltet zu haben.

<L> Lesen des eingestellten Sektors
Nach Drücken von <L> wird der eingestellte Sektor gelesen. Hierbei nützt Top-Flop den schnellen Burstmodus der neuen Commodore-Floppies 1570 und 1571. Ist der Schreib-/Lesekopf des Laufwerks schon auf der richtigen Spur positioniert, so dauert die Übertragung nur noch einen Sekundenbruchteil und ist gerade durch ein kurzes Aufleuchten des roten Lämpchens festzustellen. Aber auch Besitzer der 1541 brauchen sich nicht zu langweilen, dank Maschinensprache ist die konventionelle Übertragung immerhin auch noch recht schnell. Die nun folgenden Programmteile, die Ausgabe des Hex- und des ASCII-Dumps, sind wegen dem relativ hohen Rechenaufwand ebenfalls in Maschinensprache gehalten und gehen schnell vonstatten. Alles zusammen benötigt mit einer schnellen Floppy nur ungefähr 1 Sekunde. Das Ende des Lesevorgangs wird mit der Klingel bestätigt.

Beide Dumps (»dump« ist übrigens der Englischen Sprache entlehnt und heißt soviel wie »Abfall-, Schutthaufen«) stellen in jeder der 16 Zeilen 16 Werte dar, die zur Steigerung der Übersichtlichkeit beide in der Mitte geteilt wurden. Auch wurde auf die (seltsamerweise) sonst übliche Invertierung des ASCII-Dumps verzichtet. Die Darstellung im ASCII-Dump erfolgt gemäß dem eingestellten Anzeigemodus sowie nach einem weitverbreiteten Standard, es werden also keine Grafik- oder Steuerzeichen ausgegeben.

Lesen und Darstellen eines Sektors in 1 Sekunde

<S> Schreiben des eingestellten Sektors
Die Schreibroutine ist sehr einfach und daher kurz und schnell. Bei Eingabe des Befehls »S« werden die Daten sofort, wie sie am Bildschirm zu sehen sind, auf den eingestellten Sektor der Diskette geschrieben. Vorsicht! Dieses Kommando ist nicht mehr rückgängig zu machen. Anfängern ist also zu empfehlen, sich erst eine Sicherheitskopie der Arbeitsdiskette anzufertigen, da im schlimmsten Falle eine ganze Diskette auf diese Weise unbrauchbar gemacht werden kann. Auch ist nicht immer sicher, wie Programme auf Veränderungen mit dem Monitor reagieren; im Zweifelsfalle kann eine Hardcopy des ursprünglichen Sektors nützlich sein. Natürlich berücksichtigt die Routine aber auch den Schreibschutz auf einer Diskette, in diesem Falle wird ein »WRITE PROTECT ERROR« ausgegeben und der Schreibversuch abgebrochen.

Nach erfolgreichem Schreiben und einem Klingelzeichen befindet sich das Programm wieder im Befehlsmodus.

<E> Angezeigten Dump editieren
Der Befehl »E« für Editieren ist sehr leistungsfähig und komplex, was aus dem Listing 1 unschwer zu erkennen ist. Mit ihm ist es möglich, den dargestellten Sektor beliebig zu verändern. Der Cursor erscheint in der linken oberen Ecke des Hex-Dumps und kann mit Hilfe der Cursortasten beliebig über die beiden Dumps bewegt werden. Das Verlassen dieses Bereichs ist aus offensichtlichen Gründen nicht möglich. Um eine hexadezimale Zahl zu ändern, überschreiben Sie diese einfach. Dabei müssen allerdings immer beide Ziffern geändert werden, das Editieren nur einer der

```
---- ALLE BEFEHLE VON TOP-FLOP V2.4 ----

C = EINGABE NEUER SECTOR
D = DIRECTORY
E = ANGEZEIGTEN SECTOR EDITIEREN
F = FLOPPY-BEFEHLE
H = HARDCOPY DES ANGEZEIGTEN SECTORS
L = EINGESTELLTEN SECTOR LESEN
M = ANZEIGEMODUS MIT CRSR AENDERN
N = NAECHSTEN SECTOR LESEN
S = EINGESTELLTEN SECTOR SCHREIBEN
T = EINGABE NEUER TRACK
W = DISK INITIALISIEREN (NEUE DISK)
X = RESET ODER NEUSTART
Y = AUFRUF TEDMON ODER NEUSTART

WRITTEN 1987 BY NIKOLAUS HUBER
(C)OPYRIGHT 1987 BY MARKT & TECHNIK

ALLE RECHTE VORBEHALTEN
```

Bild 1. Ein übersichtlicher Hilfstext auf dem 40-Zeichen-Bildschirm zeigt alle gültigen Programmbefehle

beiden Ziffern wird vom Programm nicht akzeptiert. Außerdem nimmt Top-Flop im Bereich des Hex-Dumps nur die Tasten <0> bis <9> beziehungsweise <A> bis <F> an, wodurch wiederum Fehleingaben ausgeschlossen sind. Wurde ein Wert neu eingegeben, so nimmt das Programm automatisch die entsprechende Korrektur im ASCII-Dump vor, wobei dieselben Regeln für die Darstellung gelten wie beim Lesen. Der Cursor springt anschließend auf die erste Ziffer des nächsten Wertes (auch über die Spalte in der Mitte) beziehungsweise nach dem letzten Byte der Zeile an den Anfang der nächsten (in der letzten Zeile an den Anfang derselben). Da eine korrekte Eingabe mit einem Klingelzeichen bestätigt wird, können Sie ohne Probleme auch längere Wertelisten eingeben, ohne ständig den Bildschirm kontrollieren zu müssen.

Das Editieren des ASCII-Dumps läuft ganz entsprechend ab: Das Byte unter dem Cursor wird durch Überschreiben mit einem der zulässigen Zeichen (dieselben, die auch dargestellt werden können) direkt verändert und der entsprechende hexadezimale Wert sofort angeglichen. Bei einer unzulässigen Eingabe bleibt das Byte so lange auf dem ursprünglichen Wert, bis eine korrekte Taste gedrückt wird. Auch hier ist kontinuierliches Schreiben ohne weiteres möglich, da der Cursor automatisch die nächste logische Position einnimmt. Vergessen Sie aber bei allem Komfort nicht, daß die Taste <INST/DEL> keine Funktion mehr hat. Trotzdem dürfte es Ihnen nicht schwer fallen, auch längere Texte ohne große Umstände zu schreiben.

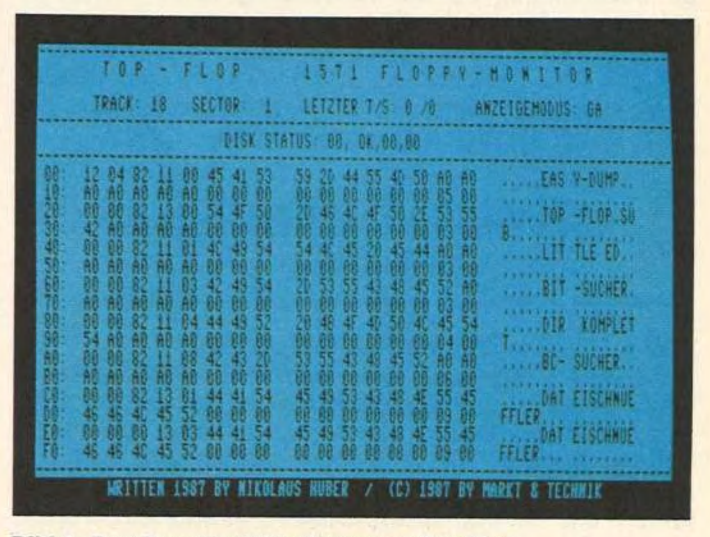


Bild 2. Der Standardbildschirm von Top-Flop

```
------------------------------------------------------------------------------
DISKETTE : PROGRAMME          NH 2A

TRACK: 18   SECTOR: 1    ANZEIGEMODUS: GA

00:  12 04 00 11 00 54 4F 50  2D 46 4C 4F 50 2E 42 41   .....TOP -FLOP.BA
10:  53 A0 A0 A0 A0 00 00 00  00 00 00 00 00 00 2F 00   S....... ......./.
20:  00 00 82 13 00 54 4F 50  2D 46 4C 4F 50 2E 53 55   .....TOP -FLOP.SU
30:  42 A0 A0 A0 A0 00 00 00  00 00 00 00 00 00 03 00   B....... ........
40:  00 00 82 13 01 45 41 53  59 2D 44 55 4D 50 A0 A0   .....EAS Y-DUMP..
50:  A0 A0 A0 A0 A0 00 00 00  00 00 00 00 00 00 05 00   ........ ........
60:  00 00 82 14 03 4C 49 54  54 4C 45 20 45 44 A0 A0   .....LIT TLE ED..
70:  A0 A0 A0 A0 A0 00 00 00  00 00 00 00 00 00 03 00   ........ ........
80:  00 00 82 13 0E 42 49 54  2D 53 55 43 48 45 52 A0   .....BIT -SUCHER.
90:  A0 A0 A0 A0 A0 00 00 00  00 00 00 00 00 00 03 00   ........ ........
A0:  00 00 82 14 01 44 49 52  20 4B 4F 4D 50 4C 45 54   .....DIR  KOMPLET
B0:  54 A0 A0 A0 A0 00 00 00  00 00 00 00 00 00 04 00   T....... ........
C0:  00 00 82 14 04 42 43 2D  53 55 43 48 45 52 A0 A0   .....BC- SUCHER..
D0:  A0 A0 A0 A0 A0 00 00 00  00 00 00 00 00 00 06 00   ........ ........
E0:  00 00 82 13 03 44 41 54  45 49 53 43 48 4E 55 45   .....DAT EISCHNUE
F0:  46 46 4C 45 52 00 00 00  00 00 00 00 00 00 09 00   FFLER... ........
------------------------------------------------------------------------------
```

Bild 3. Hardcopy eines Sektor-Inhaltes

Der Editiermodus wird mit <RETURN> beendet, worauf natürlich die Klingel ertönt, der Cursor verschwindet und der nächste Befehl eingegeben werden kann.

Übrigens werden alle Veränderungen im Editiermodus nur am Bildschirm und im Speicher durchgeführt, jedoch nicht auf der Diskette. Sie kann also mit <L> noch rückgängig gemacht werden, erst nach <S> ist die Veränderung endgültig.

<C> Eingabe der neuen Sektornummer

Mit Hilfe dieses Kommandos können Sie die aktuelle Sektornummer ändern. Nach der Eingabe von <C> (englische Schreibweise: Sector) wird die bisherige Nummer gelöscht und weiterhin in der Infozeile bei »LAST TR/SE« angezeigt. Nun sind Sie aufgefordert, die gewünschte Nummer einzugeben. Dies erfolgt übrigens in zweistellig dezimaler Form, das heißt es werden nur Zifferntasten angenommen und einstellige Zahlen müssen mit einer Null erweitert werden (zum Beispiel 05 statt 5). Dadurch sparen Sie sich die Bestätigung mit <RETURN> und somit eventuell einen dritten Tastendruck. Auch hier löscht Top-Flop unkorrekte Eingaben und quittiert mit einem Warnton. Bitte beachten Sie auch die unterschiedliche Anzahl von Sektoren auf den verschiedenen Tracks, die übrigens ebenfalls vom Programm berücksichtigt wird.

<T> Eingabe der neuen Track- und Sektornummer

Der Befehl <T> entspricht <C> mit dem Unterschied, daß hiermit auch der aktuelle Track eingegeben werden kann. Das Format ist wieder dezimal zweistellig. Sofort nach Aufruf der Routine kopiert das Programm die bisherige aktuelle Adresse an die Stelle »LAST TR/SE«. Sie können nun also jede beliebige Adresse auf der Diskette ansprechen. Im 1541- und 1570-Modus ist die höchste zulässige Tracknummer 35, bei Verwendung beider Seiten im 1571-Modus beträgt sie 70.

Auf den eingestellten aktuellen Track und Sector beziehen sich übrigens nur die Routinen »Lesen« und »Schreiben«, »Hardcopy« und »Editieren« greifen verständlicherweise auf den gerade dargestellten Sektor zurück.

Viele nützliche und leistungsfähige Befehle

<W> Diskette im Laufwerk initialisieren

Die Aufforderung für das Programm, die Diskette im Laufwerk zu initialisieren, ist eigentlich nur für den 1570- und 1571-Modus wichtig. Denn bevor auf eine Diskette mit Hilfe der Burst-Befehle zugegriffen werden kann, muß ihre ID eingelesen werden, ansonsten ist der Befehl nicht ausführbar und führt zu einem »DISK ID MISMATCH«-Fehler. Die Initialisierung wird am Anfang vom Programm automatisch durchgeführt, Sie müssen <W> also nur noch nach jedem Diskettenwechsel eingeben. Solange Sie dann nur mit einer Diskette arbeiten, ist <W> nicht mehr nötig.

<N> Nächsten Sektor der Datei laden

<N> ist wahrscheinlich das am häufigsten benötigte Kommando. Die Routine liest den nächsten Sektor der Datei, dessen Position auf der Diskette ja in den ersten beiden Bytes des letzten Sektors steht, und liest diesen ein. Dabei wird die Adresse des alten Sektors wieder in der Infozeile angezeigt. Top-Flop erkennt auch hier Sektorverkettungen, die das Ende von Dateien anzeigen (1. Byte: 00, zweites Byte: Zahl der noch benutzten Byte), und lädt in einem solchen Fall Sektor 1 auf Track 18, also den ersten Block des Inhaltsverzeichnisses nach. Die Ausgabe entspricht dann der von <L>. Mit <N> ist es theoretisch möglich, eine Datei von 100 Blöcken Länge mit einer 1570/71 innerhalb von zirka 1 Minute und 40 Sekunden komplett durchzublättern. Diese Geschwindigkeit ist ein Beispiel dafür, wie gut eingesetzte Maschinenprogramme auch innerhalb von Basicprogrammen für sehr hohes Tempo sorgen können

Vier verschiedene Anzeigemodi

<M> Aktuellen Anzeigemodus ändern

Die Möglichkeit, zwischen vier verschiedenen Anzeigemodi zu unterscheiden, ist wohl eine der komfortabelsten und auch leistungsfähigsten Optionen. Hinzu kommt, daß diese doch sehr praktische Differenzierung in keinem mir bekannten Programm dieser Art angeboten wird. Über den Befehl <M> können Sie grundsätzlich festlegen, welche Bytewerte als welche Zeichen interpretiert und dargestellt werden. Dabei sind folgende Möglichkeiten angeboten:

1. »GA«

»GA« steht für Grafikzeichensatz im ASCII-Format, das heißt die Werte von 32 bis 95 werden mit Großbuchstaben dargestellt. Dies ist die Voreinstellung.

2. »TA«

Nun kommen für den Textzeichensatz die Werte 97 bis 122 als Großbuchstaben hinzu, 65 bis 90 werden klein angezeigt. Das ASCII-Format wird vor allem zur Speicherung von Text in Basicprogrammen und zur Angabe der Dateinamen im Directory verwendet.

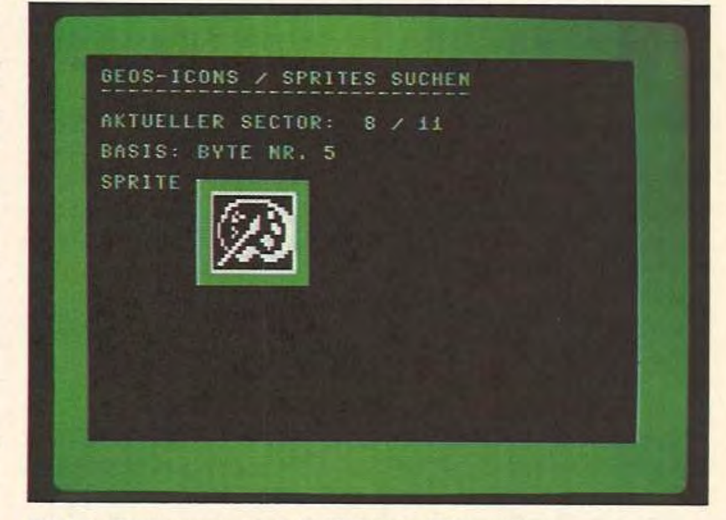


Bild 4. Spritesuche am 40-Zeichen-Bildschirm

3. »GB«

Im Bildschirm-Code-Modus legt der Computer die Texte im Bildschirmspeicher ab. Im Bildschirm-Code belegen die einfachen Buchstaben die Werte 1 bis 26, die Sonderzeichen sind dem ASCII-Code identisch.

4. »TB«
Zusätzlich werden hier noch die Werte 65 bis 90 als Großbuchstaben entsprechend TA dargestellt.
Der Bildschirmcode wird oft von Maschinenprogrammen benützt, die die Texte direkt in den Bildschirmspeicher übertragen.
Zwischen diesen vier Möglichkeiten können Sie nach Eingabe von <M> mit Hilfe der Cursortasten wählen, wobei <CURSOR abwärts> und <CURSOR rechts> bezie-

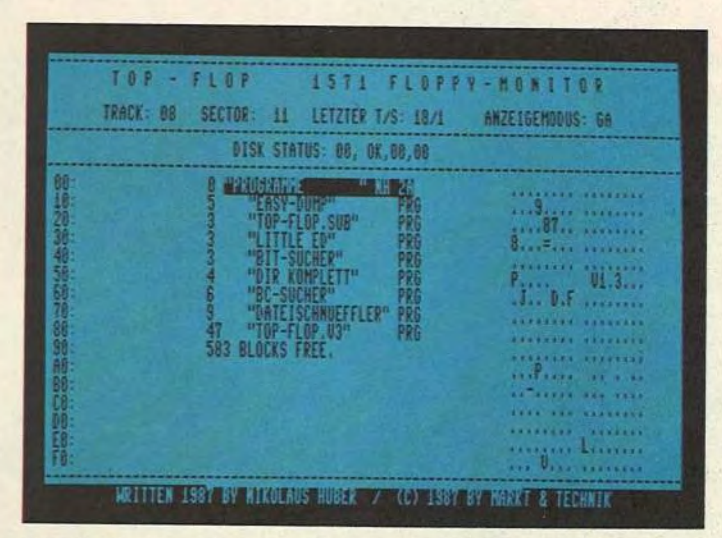


Bild 5. Die Ausgabe des Directory erfolgt in einem Fenster, das in den Bildschirm eingeblendet wird

hungsweise <CURSOR aufwärts> und <CURSOR links> die gleiche Wirkung haben. Bestätigen Sie die gewünschte Einstellung mit <RETURN>. Die Einstellung ist sowohl für das Lesen als auch für das Editieren eines Blocks wirksam.

Hardcopy auf praktisch jedem Drucker möglich

<H> Hardcopy des angezeigten Dumps
Das Erstellen einer Hardcopy des kompletten Dumps dürfte für Druckerbesitzer eine sehr nützliche Option sein, denn oft erleichert es die Arbeit ungemein, wenn mehrere Sektoren direkt verglichen werden können. Achten Sie darauf, daß bei der Eingabe von <H> der Drucker eingeschaltet und die richtige Diskette im Laufwerk ist. Andernfalls wird die Routine abgebrochen. Das Programm lädt nämlich den Namen der Diskette nach und gibt ihn zusammen mit der passenden Sektoradresse, der letzten Sektoradresse, dem aktiven Anzeigemodus sowie last not least mit dem kompletten Dump des Sektors aus (Bild 3). Um die Ausgabe zu möglichst allen Druckern kompatibel zu halten, gibt das Programm auch Buchstaben, die am Bildschirm klein geschrieben sind, als Großbuchstaben aus. Der ganze Ausdruck wird mit zwei gestrichelten Linien oben und unten begrenzt. Das Drucken selber benötigt auf einem MPS 803 nicht mehr als 35 Sekunden, wiederum dank Maschinensprache. Die Standardeinstellung des Programms schickt die Hardcopy an die Primäradresse 4, erfahrenen Programmieren dürfte es aber keine Schwierigkeiten bereiten, die Adresse im Basic- und im Maschinenprogramm ab $233A entsprechend abzuändern.

<I> Suchen nach Geos-Icons oder Sprites
Nach dem Aufruf dieser Funktion durch <I> erscheint am unteren Maskenrand die Aufforderung, auf den 40-Zeichen-Bildschirm umzuschalten. Dort werden die aktuelle Sektoradresse, das Basis-Byte, ab dem die Sprite-Information aus dem Sektor übernommen wird sowie das Sprite selber in doppelter Größe ausgegeben (Bild 4). Nun ist es möglich, das Basis-Byte mit <+> und <—> jeweils um 1 zu erhöhen beziehungsweise zu erniedrigen. Um die Suche innerhalb eines Sektors zu beschleunigen, verändern die jeweiligen Tasten zusammen mit <SHIFT> die Basis in Zehnerschritten. Nach <RETURN> stellt Top-Flop wieder den Hilfstext zur Verfügung, löscht die Aufforderung zum Umschalten und befindet sich im Normalmodus. Mit Hilfe dieser Funktion können Sie sehr leicht Spriteinformationen aus fremden Programmen heraussuchen und mit Hilfe von <Y> (siehe weiter unten) für eigene Verwendungen umkopieren.

Sprites und Icons in den Dateien suchen

<D> Ausgabe des Directory
Mit <D> wird im Arbeitsfeld das Directory der im Laufwerk liegenden Diskette ausgegeben. Dazu wird der Hexdump – allerdings nur vorübergehend – gelöscht, so daß Sie zum Beispiel nach der Anzeige eines Directory-Blockes direkt mit dem Directory vergleichen können (Bild 5). Der Einfachheit halber wurde im Programm der Basic 7.0-Befehl DIRECTORY beziehungsweise CATALOG verwendet, somit dürfte das Format keine weitere Erklärungen nötig machen. Sollte die gesamte Ausgabe mehr als 16 Zeilen lang sein, so scrollt die Liste innerhalb des Fensters. Sie können die Anzeige aber jederzeit mit <NO SCROLL> anhalten. Wenn Sie nach der Ausgabe der Anzahl der freien Blöcke eine beliebige Taste drücken, so wird intern die Leseroutine aufgerufen und der gesamte Dump nochmals dargestellt.

<F> Das Floppy-Menü
Über <F> gelangt man in ein kleines Menü mit Floppy-Befehlen, das wie das Directory auch am Platz des nun gelöschten Hex-Dumps ausgegeben wird. Sie finden dort die folgenden Menüpunkte:

<S> Scratchen (löschen) einer Datei
Nach der Eingabe des Dateinamens erfolgt eine Sicherheitsabfrage, die Sie mit <RETURN> (Voreinstellung <J> = Ausführung des Befehls) oder mit jeder anderen Taste und <RETURN> (= Rückkehr zum Menü) beantworten können.

<H> Header Disk (Diskette neu formatieren).
Zusätzlich zum Disknamen können Sie auch eingeben, ob die Diskette mit ID (Löschen aller Sektoren) oder ohne ID (nur Löschen des Inhaltsverzeichnisses) formatiert werden soll. Die ID können Sie aufgrund eines Fehlers im Basic 7.0 beim HEADER-Befehl nur mit einer Änderung direkt im Programm bestimmen. Auch hier werden Sie durch eine Sicherheitsabfrage wie bei <S> vor einer vorschnellen Fehlentscheidung bewahrt.

<V> Validate Disk (Bam neu ordnen).
Dieses Kommando wird direkt ausgeführt. Da die Floppy währenddessen nicht nutzbar ist, gibt das Programm die Meldung »BITTE WARTEN...« aus und hält den Programmablauf an, bis die Floppy wieder normal angesprochen werden kann. Nach der Ausführung jedes der drei Befehle wird der Diskstatus unter dem Text (also nicht wie normal in der speziellen Zeile) ausgegeben, was gleichzeitig das Ende der Routine und die Aufforderung zum Tastendruck signalisiert. Danach erscheint wieder das Menü.



<Z> Fenster löschen
Das Programm löscht das aktuelle Fenster und lädt den Sektor wieder nach, anschließend befindet es sich im Normalmodus.

<X> Reset oder Neustart des Programms
<X> dient zum Verlassen oder Neuinitialisieren des Programms. Bestätigen Sie die Sicherheitsabfrage nach dem Löschen des Bildschirms mit <RETURN>, so führt der Computer einen Reset aus. Beachten Sie, daß dabei eventuell von der Diskette gebootet wird, falls der Boot-Sektor beschrieben ist. Bei einer anderen Taste als <J> und <RETURN> wird das Programm über RUN neu gestartet, wonach Sie wieder auf alle Voreinstellungen zurückgreifen können.

<Y> Aufruf des TEDMON oder Neustart des Programms
Der Befehl »Y« ähnelt dem »X«-Kommando, jedoch

Der Aufbau von Top-Flop im einzelnen:

10-80	Startbildschirm	3060-3140	neue Cursorposition im Hexdump
90-150	Erkennung des Floppy- und Diskettentyps	3150-3190	neue Cursorposition im ASCII-Dump
160-240	Maschinenroutinen laden	3250-3280	aktuelle Adresse nach letzter Adresse
250-260	Hilfsbildschirm aufbauen	3290-3350	Eingabe zweistellige Dezimalzahl
270-380	Variablen initialisieren	3360-3370	auf zulässige Tracknummer prüfen
420-650	Bildschirmmaske aufbauen	3430-3470	aktuelle Adresse nach alte Adresse
660	bei 1570/71 Disk initialisieren	3480-3540	Eingabe zweistellige Dezimalzahl
700-890	Hauptmenü	3550-3580	auf zulässige Sektornummer prüfen
980-1110	Parameter für Burst-Leseroutine	3680	Anzeigemodus invertieren
1120	Burst-Leseroutine aufrufen	3690-3760	mit Cursortasten ändern
1150-1170	1541-Leseroutine	3770	Anzeigemodus normal darstellen
1220	Aufruf Hexdump	3860	Hexdump löschen
1270-1290	Dump in ASCII-Code	3870	Window für Directory definieren
1310-1330	Dump in Bildschirmcode	3880	Directory ausgeben
1490-1600	Parameter für Burst-Schreibroutine	3930-3940	Directory löschen und Windows aufheben
1610	Aufruf Burst-Schreibroutine	3950	Sektor neu lesen und darstellen
1640-1660	1541-Schreibroutine	3990-4210	Hilfstext auf Bildschirm 0 ausgeben
1740-1770	Ausgabe Diskstatus	4250-4310	Reset oder Neustart
1820-1850	aktuelle Adresse nach letzte Adresse	4370	Hexdump löschen
1860-1930	Adresse aus Sektor holen und setzen	4380-4420	Hauptmenü ausgeben
1940	Verzweigung zur Leseroutine	4430-4480	Sprungtabelle
2010-2090	Diskname lesen	4490-4570	Datei löschen
2110-2190	Infokopf für Hardcopy schreiben	4580-4670	Diskette formatieren
2210	Aufruf Hardcopy-Routine	4680-4730	BAM neu ordnen
2220-2270	untere gestrichelte Linie schreiben	4740-4760	Bereich löschen und Windows aufheben
2350-2410	Cursor auf linke obere Position setzen	4770	Sektor neu lesen und darstellen
2440-2470	Beenden des Editierens	4880-4920	TEDMON-Befehl für Dump in Tastaturpuffer
2490-2500	Auswahl Hex-/ASCII-Dump	4940-4950	Tastaturpuffer setzen
2520-2560	neue Cursorposition	4960	Aufruf Monitor
2570	Cursor ausgeben	5010-5030	Vermerk auf Bildschirm 5
2580	bei ungültiger Eingabeposition zurück	5040-5140	Ausgabe Maske auf gelöschten Bildschirm 0
2590	zur Eingabe zweier Hex-Ziffern	5150-5160	Sprite im Kasten anzeigen
2600-2620	Umwandlung in Hexzahl	5170	Spritedaten aus Sektor umkopieren
2630	Änderung im abgespeicherten Sektor	5180-5320	Auswertung der Eingabe
2650	neue Ausgabe des ASCII-Dumps	5200-5260	bei Ende Standardmasken wiederherstellen
2660-2790	Eingabe zweier Hex-Ziffern und Ausgabe	5330-5340	Ausgabe Basisbyte
2800	zurück, wenn Cursor auf Spalte	5360-5400	Editier-Unterroutine VDC-Position setzen
2840-2880	bei ungültiger Eingabe Wiederholung	5460-5510	Parameter für Burst-Diskinitialisierung
2890-2920	bei GA ohne Großbuchstaben	5520	Routine Burst-Befehl senden
2930-2970	bei ASCII direkt im Dump ausgeben	5530-5600	Parameter für Burst-Lesen setzen
2980-3050	bei BC Umwandlung und Ausgabe	5610	Burst-Lesen aufrufen



springt das Programm bei Bestätigung der Sicherheitsabfrage in den TEDMON und zeigt automatisch mit dem TEDMON-Befehl »M« den Speicherbereich von $2000 bis $20FF, in dem Top-Flop den aktuellen Sektor abgelegt hat. Sie können nun beispielsweise den Bereich disassemblieren, um Maschinenprogrammen auf die Spur zu kommen oder nach einer bestimmten Bytefolge automatisch suchen zu lassen. Der editierte Sektor kann dann folgendermaßen gespeichert werden:

```
⟨X⟩ ⟨RETURN⟩
LET TR=gewünschte Tracknummer ⟨RETURN⟩
LET SE=gewünschte Sektornummer ⟨RETURN⟩
GOTO 1430 ⟨RETURN⟩
⟨Y⟩ ⟨RETURN⟩
```

Nun befinden Sie sich wieder wie vorhin im Monitor.

Problemlose Erweiterung des Programms

Sie können das Programm natürlich dank dem modularen Aufbau ohne weiteres um eigene Routinen erweitern, denn natürlich hat jeder Anwender seine persönlichen Bedürfnisse. So könnte etwa der Einbau eines schnellen Kopierprogramms wie Tornado-Copy eine durchaus sinnvolle Erweiterung für Top-Flop sein.

Eingabe des Listings

Geben Sie zunächst das Basicprogramm (Listing 1) mit Hilfe des Checksummer 128 ein und speichern Sie es. Anschließend müssen Sie das Maschinenprogramm (Listing 2) im C 64-Modus mit dem MSE eintippen und ebenfalls sichern. Dann wird das Programm wie beschrieben gestartet.

(Nikolaus Huber(sk)

Interessante Programmiertricks in Top-Flop

- Nutzung des durch GRAPHIC 1,1 reservierten Grafikspeichers zur unproblematischen Speicherung von Daten und Maschinenprogrammen
- Ausgabe eines Hilfstextes auf dem 40-Zeichen-Bildschirm bei einem Programm mit 80-Zeichen-Darstellung
- Erkennung des Floppytypes durch Abfrage der Speicherstelle $8000 im Floppyspeicher (Zeile 90-140)
- Ausgabe einer Zahl im Hexadezimal-Format mit folgendem Leerzeichen an der aktuellen Cursorposition durch SYS DEC("B8C2"),Zahl (Zeile 630)
- Löschen der Zeile ab der aktuellen Cursorposition durch SYS DEC("C01E"),81 (Zeile 1750)
- Aufheben des definierten Windows durch SYS DEC("CA24") (Zeile 3940, 4760)
- Übergabe von Befehlen über das Programmende hinaus im Tastaturpuffer (Zeile 4890-4950)

```
10 REM * * * * * * * * * * * * * * * * INITIALISIER
   UNG * * * * * * * * * * * * * * * *                  <A99>
20 REM                                                  <AHA>
30 FAST                                                 <K2J>
40 GRAPHIC 1,1                                          <4EI>
50 PRINT "{CLR,DOWN,RVSON}* * * *{2SPACE}T O P -
   F L O P{2SPACE}V 2 . 4{2SPACE}* * * *"               <260>
60 PRINT "{DOWN}BITTE WARTEN...                         <BUJ>
70 SYS DEC("C01E"),82                                   <IEH>
80 SYS DEC("C01E"),77                                   <MI2>
90 OPEN 15,8,15                                         <SC2>
100 PRINT#15,"M-R" CHR$(0) CHR$(128) CHR$(1)            <J1E>
110 GET #15,A$: A=ASC(A$)                               <QJE>
120 IF A=117 THEN FL=70: GOTO 145                       <N8U>
130 IF A=146 THEN FL=71: GOTO 145                       <LSL>
140 FL=41                                               <70M>
145 OPEN 2,8,2,"#": PRINT#15,"U1";2;0;36;0: DCLOS
    E                                                   <A9D>
146 IF DS>0 THEN FD=0: ELSE FD=1                        <EPQ>
150 PRINT "{DOWN}DIE FLOPPY (EINHEIT 8) IST EIN 1
    "FL"{4LEFT}5{3RIGHT}LAUFWERK..."                    <NN8>
160 DOPEN #1,"TOP-FLOP.SUB"                             <MC0>
170 DCLOSE                                              <KQ7>
180 IF DS>0 THEN BEGIN                                  <N4B>
190 : PRINT "{DOWN}"DS$                                 <U3D>
200 : PRINT "{2DOWN}BITTE DISKETTE MIT TOP-FLOP.S
    UB EINLEGEN UND TASTE DRUECKEN..."                  <HGB>
210 : GET KEY G$                                        <4DB>
220 : GOTO 160                                          <3Q0>
230 BEND                                                <6DU>
240 BLOAD "TOP-FLOP.SUB"                                <N4J>
250 GOSUB 3970                                          <NJH>
260 GRAPHIC 5                                           <BBR>
270 BANK 15                                             <U7D>
280 MH$(0)="GA": MH$(1)="TA": MH$(2)="GB": MH$(3)
    ="TB": MO$="GA": RE=0: TR=18: SE=1                  <12J>
290 DIM X1(57): DIM X2(74): DIM X3(57): DIM X4(74
    )                                                   <PSQ>
300 DATA 6,58,9,59,12,60,15,61,18,62,21,63,24,64,
    27,65,32,67,35,68,38,69,41,70,44,71,47,72,50,
    73,53,74                                            <JRG>
310 DATA 58,6,59,9,60,12,61,15,62,18,63,21,64,24,
    65,27,67,32,68,35,69,38,70,41,71,44,72,47.73,
    50,74,53                                            <S81>
320 DATA 6,0,9,1,12,2,15,3,18,4,21,5,24,6,27,7,32
    ,8,35,9,38,10,41.11,44,12,47,13,50,14,53,15         <OM5>
330 DATA 58,59,60,61,62,63,64,65,67,68,69,70,71,7
    2,73,74                                             <DK0>
340 RESTORE 300                                         <ENR>
350 FOR L=0 TO 15: READ A: READ X1(A): NEXT             <U2Q>
360 FOR L=0 TO 15: READ A: READ X2(A): NEXT             <U2R>
370 FOR L=0 TO 15: READ A: READ X3(A): NEXT             <U2R>
380 FOR L=0 TO 15: READ A: X4(A)=L: NEXT                <FSF>
390 REM                                                 <HQ7>
400 REM * * * * * * * * * * * * * * *BILDSCHIRMMA
    SKE* * * * * * * * * * * * * * *                    <G20>
410 REM                                                 <GA1>
420 SCNCLR                                              <PBJ>
430 PRINT "---------------------------------------
    ----------------------------------------"           <PIB>
440 PRINT "{8SPACE}T O P{2SPACE}-{2SPACE}F L O P{
    8SPACE}1 5{7SPACE}F L O P P Y - M O N I T O R
    "                                                   <3SG>
450 PRINT "{DOWN,7SPACE}TRACK:{6SPACE}SECTOR:{7SP
    ACE}LETZTER T/S:{3SPACE}/{6SPACE}ANZEIGEMODUS
    :                                                   <TGH>
460 PRINT "---------------------------------------
    ----------------------------------------"           <P6B>
470 PRINT "{23SPACE}DISK STATUS:                        <HRL>
480 PRINT "---------------------------------------
    ----------------------------------------"           <OUB>
490 GOSUB 1720                                          <JR0>
500 SYS DEC("C00C"),2                                   <83C>
510 CHAR ,0,23,"----------------------------------
    -----------------------------------------
    --"                                                 <E3I>
520 CHAR ,0,24,"{9SPACE}WRITTEN 1987 BY NIKOLAUS
    HUBER{2SPACE}/{2SPACE}(C) 1987 BY MARKT & TEC
    HNIK{9SPACE}",1                                     <U0G>
530 SYS DEC("C01E"),79                                  <7S7>
540 CHAR ,37,1,MID$(STR$(FL),2,1)                       <3AH>
550 CHAR ,39,1,RIGHT$(STR$(FL),1)                       <05Q>
560 CHAR ,13,3,STR$(TR)                                 <0J0>
570 CHAR ,27,3,STR$(SE)                                 <5RT>
580 CHAR ,46,3,"0"                                      <SE7>
590 CHAR ,49,3,"0"                                      <C17>
600 CHAR ,69,3,MH$(0)                                   <KRF>
610 FOR L=0 TO 15                                       <NTK>
620 : CHAR ,1,(7+L)                                     <43Q>
630 : SYS DEC("B8C2"),L*16                              <PTC>
640 : SYS DEC("C00C"),58                                <P7T>
650 NEXT                                                <LEG>
660 IF FL<>41 THEN 5440                                 <L3E>
670 REM                                                 <GQ3>
680 REM * * * * * * * * * * * * * * HAUPTMENUE* * *
     * * * * * * * * * * * * * * *                      <J2C>
690 REM                                                 <JAD>
700 SYS DEC("C01E"),77                                  <5KE>
710 FG=0                                                <HM3>
720 SLOW                                                <TBG>
730 GET KEY G$                                          <KCB>
740 IF G$="L" THEN 930                                  <T37>
750 IF G$="S" THEN 1430                                 <GVR>
760 IF G$="N" THEN 1790                                 <KFK>
770 IF G$="H" THEN 1960                                 <HF6>
780 IF G$="E" THEN 2310                                 <M7M>
790 IF G$="T" THEN 3210                                 <JNH>
800 IF G$="C" THEN 3390                                 <H7C>
810 IF G$="M" THEN 3640                                 <J7A>
820 IF G$="D" THEN 3810                                 <NVG>
830 IF G$="X" THEN 4230                                 <I7R>
840 IF G$="F" THEN 4330                                 <KVU>
850 IF G$="Y" THEN 4790                                 <JVT>
860 IF G$="I" THEN 4970                                 <JNC>
870 IF G$="W" THEN 5440                                 <J7G>
880 SOUND 1,2000,19                                     <D77>
890 GOTO 700                                            <RAA>
900 REM                                                 <MIQ>
910 REM * * * * * * * * * * * * * * LESEN & AUSGEBE
    N* * * * * * * * * * * * * * * *                    <MQR>
920 REM                                                 <M2D>
930 SYS DEC("C98E")                                     <PVK>
940 FAST                                                <1O3>
950 RE=1                                                <4SN>
```

```
960 OPEN 1,8,15                                          <RV2>
970 IF FL=41 THEN 1150                                   <1SB>
980 POKE DEC("FF00"),0                                   <NTV>
990 A=DEC("1430")                                        <7J2>
1000 REM POKEA+0,85                                      <34C>
1010 REM POKEA+1,48                                      <3CD>
1020 REM POKEA+2,4                                       <3KE>
1030 REM SYSDEC("2200"),,,3                              <3SF>
1040 POKE A+2,0                                          <885>
1050 POKE A+3,TR                                         <3Q3>
1060 POKE A+4,SE                                         <4UJ>
1070 POKE A+5,1                                          <ACB>
1080 POKE DEC("00B0"),1                                  <1NJ>
1090 POKE DEC("00B1"),255                                <6UE>
1100 POKE DEC("00FB"),0                                  <GDN>
1110 POKE DEC("00FC"),32                                 <3F0>
1120 SYS DEC("2220"),,,6                                 <BJ2>
1130 DCLOSE                                              <DQ7>
1140 GOTO 1190                                           <2GV>
1150 OPEN 5,8,5,"#"                                      <DDS>
1160 PRINT#1,"U1";5;0;TR;SE                              <FNR>
1170 SYS DEC("24D0")                                     <DT2>
1180 DCLOSE                                              <8C7>
1190 GOSUB 1720: IF DS<>0 THEN 700                       <IP6>
1200 BANK 15                                             <21D>
1210 CHAR ,6,7                                           <FDB>
1220 SYS DEC("23A7")                                     <L80>
1230 CHAR ,58,7                                          <MVI>
1240 IF LEFT$(MO$,1)="T" THEN SYS DEC("C00C"),14:
      POKE DEC("C0"),1                                   <ME4>
1250 : ELSE POKE DEC("C0"),0                             <HML>
1260 IF RIGHT$(MO$,1)="B" THEN 1310                      <ESB>
1270 POKE DEC("84"),0                                    <664>
1280 POKE DEC("85"),32                                   <D0C>
1290 SYS DEC("23DE")                                     <PRI>
1300 GOTO 1350                                           <0GN>
1310 POKE DEC("84"),0                                    <6M6>
1320 POKE DEC("85"),32                                   <F04>
1330 SYS DEC("2448")                                     <PJI>
1340 SYS DEC("C00C"),0                                   <RD1>
1350 SYS DEC("C00C"),13                                  <GAO>
1355 SYS DEC("C00C"),142                                 <5L4>
1360 T=0                                                 <QFD>
1370 SYS DEC("C98E")                                     <KM1>
1380 IF FG=1 THEN 3130                                   <2S2>
1390 GOTO 700                                            <12A>
1400 REM                                                 <144>
1410 REM * * * * * * * * * * * * *SCHREIBEN* * *
      * * * * * * * * * * * * * * * *                    <1C5>
1420 REM                                                 <1K6>
1430 SYS DEC("C98E")                                     <K23>
1440 FAST                                                <QN3>
1450 IF RE=0 THEN 880                                    <NAJ>
1460 OPEN 1,8,15                                         <571>
1470 IF FL=41 THEN 1630                                  <D7C>
1480 POKE DEC("FF00"),0                                  <DG3>
1490 A=DEC("1430")                                       <HFP>
1500 POKE A+0,85                                         <D4G>
1510 POKE A+1,48                                         <QGU>
1520 POKE A+2,2                                          <E04>
1530 POKE A+3,TR                                         <FQ3>
1540 POKE A+4,SE                                         <CUJ>
1550 POKE A+5,1                                          <ECB>
1560 POKE DEC("00B0"),1                                  <3NR>
1570 POKE DEC("00B1"),255                                <EVE>
1580 POKE DEC("00FB"),0                                  <GDN>
1590 POKE DEC("00FC"),32                                 <3F0>
1600 SYS DEC("228E"),,,6                                 <CJI>
1610 DCLOSE                                              <D27>
1620 GOTO 1660                                           <0OU>
1630 OPEN 5,8,5,"#"                                      <D5T>
1640 SYS DEC("24E4")                                     <U5D>
1650 PRINT#1,"U2";5;0;TR;SE                              <SDR>
1655 DCLOSE                                              <GF7>
1660 GOSUB 1720                                          <K71>
1670 PRINT CHR$(7)                                       <USQ>
1680 GOTO 700                                            <0EA>
1690 REM                                                 <2C9>
1700 REM * * * * * * * * * * * * * *AUSGABE DISKSTA
      TUS * * * * * * * * * * * * * * * *                <0K2>
1710 REM                                                 <0S3>
1720 CHAR ,36,5                                          <I7N>
1730 SYS DEC("C01E"),81                                  <58C>
1740 CHAR ,36,5,DS$                                      <D82>
1750 RETURN                                              <D27>
1760 REM                                                 <144>
1770 REM * * * * * * * * * * * * * * NAECHSTER SECT
      OR* * * * * * * * * * * * * * * *                  <1C5>
1780 REM                                                 <2KA>
1790 FAST                                                <MIJ>
1800 LT=TR                                               <9TN>
1810 LS=SE                                               <5D7>
1820 CHAR ,45,3,STR$(LT)+"{5SPACE}"                      <4P0>
1830 CHAR ,48,3,STR$(LS)                                 <M1T>
1840 TR=PEEK(DEC("2000"))                                <N45>
1850 SE=PEEK(DEC("2001"))                                <UL3>
1860 CHAR ,48,3,"/"                                      <D82>
1870 IF TR<1 OR TR>70 OR SE>20 THEN TR=18: SE=1          <S39>
1880 CHAR ,13,3,"{3SPACE}"                               <MG4>
1890 CHAR ,27,3,"{3SPACE}"                               <MS4>
1900 CHAR ,13,3,STR$(TR)                                 <Q88>
1910 CHAR ,27,3,STR$(SE)                                 <V0L>
1920 GOTO 930                                            <56L>
1930 REM                                                 <7CT>
1940 REM * * * * * * * * * * * * * *HARDCOPY * * *
      * * * * * * * * * * * * * * * * *                  <6KQ>
1950 REM                                                 <6SR>
1960 SYS DEC("C98E")                                     <5U4>
1970 FAST                                                <URJ>
1980 IF RE=0 THEN 880                                    <FUH>
1990 OPEN 1,8,15: OPEN 2,8,2,"#"                         <R6E>
2000 PRINT#1,"U1";2;0;18;0                               <FGT>
2010 PRINT#1,"B-P";2;144                                 <2G4>
2020 B$=""                                               <6S9>
2030 FOR L=1 TO 24                                       <JPS>
2040 : GET #2,A$                                         <SNQ>
2050 : B$=B$+A$                                          <LIK>
2060 NEXT                                                <AD0>
2070 DCLOSE                                              <DE7>
2080 GOSUB 1720: IF DS<>0 THEN 700                       <H16>
2090 OPEN 4,4                                            <P9E>
2100 FOR L=1 TO 80                                       <GPK>
2110 : PRINT#4,"-";                                      <51T>
2120 NEXT L                                              <DD6>
2130 PRINT#4," DISKETTE : ";B$                           <1RH>
2140 PRINT#4                                             <3M1>
2150 PRINT#4," TRACK:";TR;"{2SPACE}SECTOR:";SE;"{
      3SPACE}ANZEIGEMODUS: ";MO$                         <H1F>
2160 PRINT#4                                             <3U1>
2170 CLOSE 4                                             <SUL>
2180 BANK 15                                             <25D>
2190 SYS DEC("233A")                                     <C14>
2200 OPEN 4,4                                            <DTE>
2210 FOR L=1 TO 80                                       <NPK>
2220 : PRINT#4,"-";                                      <D5T>
2230 NEXT L                                              <LA6>
2240 PRINT#4                                             <3E1>
2250 CLOSE 4                                             <SML>
2260 PRINT CHR$(7)                                       <ASQ>
2270 GOTO 700                                            <12A>
2280 REM                                                 <34C>
2290 REM * * * * * * * * * * * * * *EDITIEREN* * *
      * * * * * * * * * * *.* * * * * *                  <3CD>
2300 REM                                                 <1K6>
2310 SYS DEC("C98E")                                     <K23>
2320 FAST                                                <AL3>
2330 P=6                                                 <ALE>
2340 Q=7                                                 <40K>
2350 IF RE=0 THEN 880                                    <7AJ>
2360 AD=P+80*Q                                           <IBR>
2370 GOSUB 5340: SYS DEC("CDDA"),,31: RREG X             <MQ3>
2380 Y=X+128                                             <7EG>
2390 GOSUB 5340: SYS DEC("CDCC"),Y,31                    <9DN>
2400 GET KEY G$                                          <70B>
2410 IF G$="{DOWN}" OR G$="{RIGHT}" OR G$="{UP}"
      OR G$="{LEFT}" THEN 2500                           <EHA>
2420 IF G$=CHR$(13) THEN BEGIN                           <3MN>
2430 : GOSUB 5340: SYS DEC("CDCC"),X,31                  <483>
2440 : PRINT CHR$(7)                                     <GS2>
2450 : GOTO 700                                          <HMA>
2460 BEND                                                <10E>
2470 IF P<58 THEN 2560                                   <DS8>
2480 IF P>57 THEN 2780                                   <AC1>
2490 GOTO 2400                                           <3GG>
2500 GOSUB 5340: SYS DEC("CDCC"),X,31:                   <7RR>
2510 IF G$="{DOWN}" THEN Q=Q+1: IF Q=23 THEN Q=22        <8JN>
2520 IF G$="{UP}" THEN Q=Q-1: IF Q=6 THEN Q=7            <NB4>
2530 IF G$="{RIGHT}" THEN P=P+1: IF P=75 THEN P=7
      4                                                  <ABI>
2540 IF G$="{LEFT}" THEN P=P-1: IF P=5 THEN P=6          <ND6>
2550 GOTO 2360                                           <20O>
2560 IF X1(P)=0 THEN 2400                                <SCD>
2570 GOSUB 2640                                          <JN9>
2580 IF X<7 THEN X=X+64                                  <JTS>
2590 IF B<7 THEN B=B+64                                  <LJK>
2600 HE=DEC(CHR$(X)+CHR$(B))                             <2PH>
2610 POKE DEC("2000")+X3(P)+(Q-7)*16,HE                  <5UT>
2620 FG=1                                                <MMJ>
2630 GOTO 1230                                           <0OJ>
2640 GOSUB 5340: SYS DEC("CDCC"),32,31                   <1B2>
2650 AF=1: GOSUB 5340: SYS DEC("CDCC"),32,31             <EIP>
2660 IF G$>"/" AND G$<":" OR G$>"@" AND G$<"G" TH
      EN 2680                                            <3AM>
2670 GOTO 2400                                           <10G>
2680 X=ASC(G$)                                           <SUN>
2690 IF X>64 THEN X=X-64                                 <I35>
2700 GOSUB 5340: SYS DEC("CDCC"),X,31                    <0DJ>
2710 GET KEY G$                                          <6OB>
2720 IF G$>"/" AND G$<":" OR G$>"@" AND G$<"G" TH
      EN 2740                                            <6QO>
2730 GOTO 2710                                           <18N>
2740 B=ASC(G$)                                           <OMN>
2750 IF B>64 THEN B=B-64                                 <BH2>
2760 AF=1: GOSUB 5340: SYS DEC("CDCC"),B,31              <CK2>
2770 RETURN                                              <OU7>
2780 IF P=66 THEN GOTO 2400                              <FBI>
2790 GOSUB 5340: SYS DEC("CDCC"),32,31                   <HN1>
2800 SYS DEC("C98E")                                     <5E6>
2810 B=ASC(G$)                                           <L6N>
2820 IF B<32 OR B>90 AND B<97 OR B>122 AND B<193
      OR B>218 THEN BEGIN                                <M3R>
2830 : SOUND 1,2000,19                                   <BHT>
2840 : GET KEY G$                                        <84A>
2850 : GOTO 2810                                         <848>
2860 BEND                                                <1KE>
2870 IF LEFT$(MO$,1)="G" AND B>90 THEN BEGIN             <DDE>
```

Listing 1. Das Hauptprogramm »TOP-FLOP« in Basic. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
2880 : B=255                                         <A4F>
2890 : GOTO 2820                                     <ESB>
2900 BEND                                            <HME>
2910 IF RIGHT$(MO$,1)="A" THEN BEGIN                 <K35>
2920 : CHAR ,X2(P),Q                                 <GV4>
2930 : SYS DEC("B8A5"),B                             <0E9>
2940 : POKE DEC("2000")+X4(P)+16*(Q-7),B             <KTO>
2950 BEND                                            <5KU>
2960 IF B>63 AND B<91 THEN B=B-64                    <43V>
2970 IF B>96 AND B<123 THEN B=B-32                   <LCL>
2980 IF B>192 THEN B=B-96-32                         <HVE>
2990 IF RIGHT$(MO$,1)="B" THEN BEGIN                 <LN5>
3000 : CHAR ,X2(P),Q                                 <GB6>
3010 : SYS DEC("B8A5"),B                             <AF1>
3020 : POKE DEC("2000")+X4(P)+16*(Q-7),B             <SSO>
3030 BEND                                            <5SU>
3040 GOSUB 5340: SYS DEC("CDCC"),B,31                <A2Q>
3050 P=P+1                                           <LA7>
3060 IF P=75 THEN BEGIN                              <CNA>
3070 : P=58                                          <CT3>
3080 : Q=Q+1                                         <RUD>
3090 : IF Q=23 THEN Q=22                             <7FJ>
3100 BEND                                            <1SE>
3110 IF P=66 THEN P=67                               <VCH>
3120 GOTO 2360                                       <0OD>
3130 P=P+3                                           <L66>
3140 IF P=30 THEN P=32                               <FBN>
3150 IF P=56 THEN P=6: Q=Q+1                         <G10>
3160 IF Q=23 THEN Q=22                               <F0I>
3170 GOTO 2360                                       <1GD>
3180 REM                                             <3KE>
3190 REM * * * * * * * * * * * * *EINGABE TRACK*
     * * * * * * * * * * * * * * * * *               <3SF>
3200 REM                                             <040>
3210 SYS DEC("C98E")                                 <KE2>
3220 FAST                                            <IM3>
3230 LT=TR                                           <H2N>
3240 CHAR ,48,3,STR$(LS)                             <D2H>
3250 CHAR ,48,3,"/"                                  <LK1>
3260 CHAR ,14,3,"{3SPACE}"                           <K4D>
3270 GET KEY G$                                      <4OB>
3280 IF G$<"0" OR G$>"9" THEN 3270                   <DRU>
3290 CHAR ,14,3,G$                                   <8UB>
3300 GET KEY H$                                      <684>
3310 IF H$<"0" OR G$>"9" THEN 3300                   <ICB>
3320 CHAR ,15,3,H$                                   <B2N>
3330 TR=10*VAL(G$)+VAL(H$)                           <LND>
3340 IF FD=0 THEN IF TR<1 OR TR>35 THEN SOUND 1,2
     000,19: GOTO 3260                               <N78>
3350 IF FD=1 THEN IF TR<1 OR TR>70 THEN SOUND 1,2
     000,19: GOTO 3260                               <P7C>
3360 REM                                             <144>
3370 REM * * * * * * * * * * * * *EINGABE SECTOR
     * * * * * * * * * * * * * * * * *               <1C5>
3380 REM                                             <2KA>
3390 SYS DEC("C98E")                                 <KQ0>
3400 FAST                                            <AH3>
3410 LS=SE                                           <587>
3420 CHAR ,45,3,STR$(LT)+"{4SPACE}"                  <80P>
3430 CHAR ,48,3,STR$(LS)                             <63T>
3440 CHAR ,48,3,"/"                                  <TO0>
3450 CHAR ,28,3,"{4SPACE}"                           <DS7>
3460 GET KEY G$                                      <7GB>
3470 IF G$<"0" OR G$>"9" THEN 3460                   <EBJ>
3480 CHAR ,28,3,G$                                   <KV8>
3490 GET KEY H$                                      <764>
3500 IF H$<"0" OR H$>"9" THEN 3490                   <COC>
3510 CHAR ,29,3,H$                                   <V3N>
3520 SE=10*VAL(G$)+VAL(H$)                           <4P9>
3530 IF TR>=1 AND TR<=17 OR TR>=36 AND TR<=52 THE
     N IF SE>20 THEN 3590                            <4P3>
3540 IF TR>=18 AND TR<=24 OR TR>=53 AND TR<=59 TH
     EN IF SE>18 THEN 3590                           <OIM>
3550 IF TR>=25 AND TR<=30 OR TR>=60 AND TR<=65 TH
     EN IF SE>17 THEN 3590                           <HBE>
3560 IF TR>=31 AND TR<=35 OR TR<=66 AND TR>=70 TH
     EN IF SE>16 THEN 3590                           <0BQ>
3570 SYS DEC("C98E")                                 <KU0>
3580 GOTO 700                                        <1MA>
3590 SOUND 1,2000,19                                 <GAI>
3600 GOTO 3450                                       <1GH>
3610 REM                                             <2C9>
3620 REM * * * * * * * * * * * * *ANZEIGEMODUS *
     * * * * * * * * * * * * * * * * *               <2KA>
3630 REM                                             <2SB>
3640 SYS DEC("C98E")                                 <4M1>
3650 FAST                                            <EHJ>
3660 CHAR ,69,3,MO$,1                                <7I0>
3670 GET KEY G$                                      <6DB>
3680 IF G$="{UP}" OR G$="{LEFT}" THEN MH=MH-1: IF
      MH=-1 THEN MH=3                                <18Q>
3690 IF G$="{DOWN}" OR G$="{RIGHT}" THEN MH=MH+1:
      IF MH=4 THEN MH=0                              <OJT>
3700 IF G$="{DOWN}" OR G$="{UP}" OR G$="{RIGHT}"
     OR G$="{LEFT}" THEN BEGIN                       <05T>
3710 : MO$=MH$(MH)                                   <CHE>
3720 : CHAR ,69,3,MO$,1                              <NO1>
3730 : GOTO 3670                                     <1KM>
3740 BEND                                            <1OE>
3750 CHAR ,69,3,MO$                                  <2KT>
3760 SYS DEC("C98E")                                 <4M1>
3770 GOTO 700                                        <0QA>
3780 REM                                             <0K2>
3790 REM * * * * * * * * * * * * *DIRECTORY* * *
     * * * * * * * * * * * * * * * * *               <0S3>
3800 REM                                             <54K>
3810 SYS DEC("C98E")                                 <L67>
3820 FAST                                            <2S3>
3830 SYS DEC("C01E"),76                              <DDC>
3840 WINDOW 6,7,55,22,1                              <BGV>
3850 WINDOW 20,7,55,22                               <CP5>
3860 DIRECTORY                                       <O3P>
3870 SYS DEC("C01E"),77                              <F94>
3880 SYS DEC("CA24")                                 <7UF>
3890 GOSUB 1720: IF DS<>0 THEN GOTO 700              <D65>
3900 GET KEY G$                                      <0GB>
3910 WINDOW 20,7,55,22,1                             <O1Q>
3920 SYS DEC("CA24")                                 <7ED>
3930 GOTO 930                                        <42L>
3940 REM                                             <4KI>
3950 REM * * * * * * * * * * * * *HILFSTEXT* * *
     * * * * * * * * * * * * * * * * *               <4SJ>
3960 REM                                             <44G>
3970 GRAPHIC 0,1                                     <PAI>
3980 SYS DEC("C01E"),77                              <9PU>
3990 PRINT "---------------------------------------
     ---"                                            <229>
4000 PRINT "---- ALLE BEFEHLE VON TOP-FLOP V2.4 -
     ---"                                            <GTK>
4010 PRINT "---------------------------------------
     ---"                                            <2QA>
4020 PRINT "{DOWN} C = EINGABE NEUER SE{RVSON}C{R
     VOFF}TOR                                        <J6C>
4030 PRINT " D = {RVSON}D{RVOFF}IRECTOY              <HTO>
4040 PRINT " E = ANGEZEIGTEN SECTOR {RVSON}E{RVOF
     F}DITIEREN                                      <62N>
4050 PRINT " F = {RVSON}F{RVOFF}LOPPY-BEFEHLE        <76T>
4060 PRINT " H = {RVSON}H{RVOFF}ARDCOPY DES ANGEZ
     EIGTEN SECTORS                                  <QSV>
4070 PRINT " L = EINGESTELLTEN SECTOR {RVSON}L{RV
     OFF}ESEN                                        <B8P>
4080 PRINT " M = ANZEIGE{RVSON}M{RVOFF}ODUS MIT C
     RSR AENDERN                                     <9S0>
4090 PRINT " N = {RVSON}N{RVOFF}AECHSTEN SECTOR L
     ESEN                                            <TJN>
4100 PRINT " S = EINGESTELLTEN SECTOR {RVSON}S{RV
     OFF}CHREIBEN                                    <4FD>
4110 PRINT " T = EINGABE NEUER {RVSON}T{RVOFF}RAC
     K                                               <I7G>
4120 PRINT " W = DISK INITIALISIEREN (NE{RVSON}W{
     RVOFF}DISK)                                     <707>
4130 PRINT " X = RESET ODER NEUSTART                 <22O>
4140 PRINT " Y = AUFRUF TEDMON ODER NEUSTART         <TMC>
4150 PRINT "{2DOWN}--------------------------------
     -----------"                                    <HG7>
4160 PRINT "{5SPACE}WRITTEN 1987 BY NIKOLAUS HUBE
                                                     <88D>
4170 PRINT "{3SPACE}(C)OPYRIGHT 1987 BY MARKT & T
     ECHNIK                                          <4IC>
4180 PRINT "{DOWN,8SPACE}ALLE RECHTE VORBEHALTEN     <RK6>
4190 PRINT "---------------------------------------
     ---"                                            <3IF>
4200 RETURN                                          <9K7>
4210 REM * * * * * * * * * * * * *PROGRAMM BEENDE
     N * * * * * * * * * * * * * * * * *             <7CT>
4220 REM                                             <7KU>
4230 SYS DEC("C98E")                                 <LI5>
4240 FAST                                            <QR3>
4250 SCNCLR                                          <4TJ>
4260 PRINT "{DOWN,RIGHT}'TOP FLOP'{2SPACE}BEENDEN
     ...                                             <A14>
4270 INPUT "{DOWN,RIGHT}SIND SIE SICHER{3RIGHT}J{
     3LEFT}";A$                                      <20F>
4280 IF A$="J" THEN SYS DEC("FF3D")                  <7V2>
4290 RUN                                             <UJU>
4300 REM                                             <7KU>
4310 REM * * * * * * * * * * * * *FLOPPY-BEFEHLE
     * * * * * * * * * * * * * * * * *               <7SV>
4320 REM                                             <74S>
4330 SYS DEC("C98E")                                 <LM5>
4340 FAST                                            <IQ3>
4350 WINDOW 6,7,55,22,1                              <E09>
4360 PRINT TAB(10)"{DOWN}FLOPPY-BEFEHLE:             <OAF>
4370 PRINT TAB(10) "{DOWN}'S'CRATCH (DATEI LOESCH
     EN)                                             <QOD>
4380 PRINT TAB(10)"'H'EADER (DISKETTE FORMATIEREN
     )                                               <8QM>
4390 PRINT TAB(10) "'V'ALIDATE (BAM NEU ORDNEN)      <11J>
4400 PRINT TAB(10)"'Z'URUECK ZUM MONITOR             <1SN>
4410 GET KEY G$                                      <18B>
4420 IF G$="S" THEN 4470                             <0HQ>
4430 IF G$="H" THEN 4560                             <6PO>
4440 IF G$="V" THEN 4660                             <11P>
4450 IF G$="Z" THEN 4720                             <3PU>
4460 GOTO 4410                                       <009>
4470 SYS DEC("C98E")                                 <L27>
4480 INPUT "{DOWN}DATEINAME ";A$                     <RTV>
4490 IF LEN(A$)>16 THEN 4480                         <BBT>
4500 INPUT "{DOWN}SIND SIE SICHER{3RIGHT}J{3LEFT}
     ";SU$                                           <SML>
4510 IF SU$<>"J" THEN 4350                           <FS0>
4520 SCRATCH(A$)                                     <TH6>
4530 PRINT "{DOWN}"DS$                               <A3D>
4540 GET KEY G$                                      <0GB>
4550 GOTO 4350                                       <2O3>
4560 SYS DEC("C98E")                                 <567>
4570 INPUT "{DOWN}DISKNAME ";A$                      <5KS>
4580 IF LEN(A$)>16 THEN 4580                         <7QF>
4590 INPUT "{DOWN}MIT ID FORMATIEREN{3RIGHT}J{3LE
     FT}";B$                                         <DFA>
```

```
4600 INPUT "{DOWN}SIND SIE SICHER{3RIGHT}J{3LEFT}
     ";SU$                                          <QMD>
4610 IF SU$<>"J" THEN 4350                          <9SO>
4620 IF B$="J" THEN HEADER(A$),INH: ELSE HEADER(A
     $)                                             <TI2>
4630 PRINT "{DOWN}"DS$                              <63D>
4640 GET KEY G$                                     <10B>
4650 GOTO 4350                                      <383>
4660 SYS DEC("C98E")                                <5A6>
4670 PRINT "{DOWN}BITTE WARTEN..."                  <25M>
4680 COLLECT                                        <OH3>
4690 PRINT "{DOWN}"DS$                              <I3D>
4700 GET KEY G$                                     <0GB>
4710 GOTO 4350                                      <2O3>
4720 SYS DEC("C98E")                                <567>
4730 WINDOW 6,7,55,22,1                             <90L>
4740 SYS DEC("CA24")                                <72C>
4750 GOTO 960                                       <5IG>
4760 REM                                            <44G>
4770 REM * * * * * * * * * * * * *PROGRAMM BEENDE
     N UND TEDMON AUFRUFEN * * * * *                <4CH>
4780 REM                                            <7KU>
4790 SYS DEC("C98E")                                <LI5>
4800 SCNCLR                                         <LVJ>
4810 PRINT "{DOWN,RIGHT}'TOP-FLOP' BEENDEN UND TE
     DMON AUFRUFEN...                               <JR9>
4820 INPUT "{DOWN,RIGHT}SIND SIE SICHER{3RIGHT}J{
     3LEFT}";G$                                     <IAD>
4830 IF G$<>"J" THEN RUN                            <36I>
4840 SCNCLR                                         <LNJ>
4850 SYS DEC("C01E"),76                             <0CO>
4860 RESTORE 4910                                   <H37>
4870 FOR L=0 TO 13                                  <TTK>
4880 : READ X                                       <RME>
4890 : POKE DEC("034A")+L,X                         <6IH>
4900 NEXT                                           <QB0>
4910 DATA 77,32,70,50,48,48,48,32,70,50,48,70,70,
     13                                             <KVU>
4920 POKE DEC("A20"),14                             <372>
4930 POKE DEC("D0"),14                              <OUK>
4940 MONITOR                                        <0UA>
4950 REM                                            <3SF>
4960 REM * * * * * * * * * * * * *SPRITES SUCHEN
     * * * * * * * * * * * * * * * *                <34C>
4970 SYS DEC("C98E")                                <KM1>
4980 BB=0                                           <049>
4990 CHAR ,20,23                                    <4R1>
5000 SYS DEC("C00C"),15                             <P3T>
5010 PRINT "{RVSON}BITTE AUF 40-ZEICHEN-MONITOR U
     MSCHALTEN"                                     <9D7>
5020 GRAPHIC 0,1                                    <MAI>
5030 PRINT "{DOWN,RIGHT}GEOS-ICONS / SPRITES SUCH
     EN                                             <UCH>
5040 PRINT "{RIGHT}----------------------------     <G8B>
5050 PRINT "{DOWN,RIGHT}AKTUELLER SECTOR: "TR"/"S
     E                                              <N2B>
5060 PRINT "{DOWN,RIGHT}BASIS: BYTE NR."BB          <4NO>
5070 PRINT "{DOWN,RIGHT}SPRITE                      <S6S>
5080 CHAR ,8,8,"YYYYYYYY",1                         <VHI>
5090 FOR L=1 TO 5                                   <18G>
5100 : CHAR ,8,8+L," {6RIGHT} ",1                   <SFG>
5110 NEXT                                           <E1G>
5120 CHAR ,8,14," YYYYYY ",1                        <HNN>
5130 SPRITE 1,1,2,,1,1                              <FB3>
5140 MOVSPR 1,96,122                                <75O>
5150 SYS DEC("24BF"),,BB                            <TTA>
5160 GET KEY G$                                     <30B>
5170 IF G$<>"+" AND G$<>"-" AND G$<>"±" AND G$<>"
     =" AND G$<>CHR$(13) THEN 5160                  <B77>
5180 IF G$=CHR$(13) THEN BEGIN                      <2EN>
5190 : GOSUB 3970                                   <A7J>
5200 : SPRITE 1,0                                   <MLG>
5210 : GRAPHIC 5                                    <IBQ>
5220 : CHAR ,20,23,"------------------------------
     -----------"                                   <1MT>
5230 : GOTO 700                                     <K6A>
5240 BEND                                           <9HE>
5250 IF G$="+" THEN BB=BB+1                         <B55>
5260 IF G$="±" THEN BB=BB+10                        <VJC>
5270 IF G$="-" THEN BB=BB-1                         <2LJ>
5280 IF G$="=" THEN BB=BB-10                        <SK0>
5290 IF BB>255 THEN BB=255                          <MDR>
5300 IF BB<0 THEN BB=0                              <EP8>
5310 CHAR ,16,6,"{5SPACE}"                          <ILO>
5320 CHAR ,16,6,STR$(BB)                            <N1C>
5330 GOTO 5150                                      <183>
5340 IF AF=0 THEN AH=AD: ELSE AH=AD+1               <V34>
5350 AF=0                                           <LAT>
5360 SYS DEC("CDCC"),AH/256,18                      <BG0>
5370 SYS DEC("CDCC"),AH AND 255,19                  <9TT>
5380 RETURN                                         <987>
5390 REM                                            <4SJ>
5400 REM * * * * * * * * * * * * DISKETTE INITIAL
     ISIEREN * * * * * * * * * * * * *              <54K>
5410 REM                                            <5CL>
5420 SYS DEC("C98E")                                <527>
5430 FAST                                           <6SJ>
5440 A=DEC("1430")                                  <PJQ>
5450 POKE DEC("FF00"),0                             <M1P>
5460 OPEN 1,8,15                                    <L71>
5470 POKE A,85                                      <3O6>
5480 POKE A+1,48                                    <1GU>
5490 POKE A+2,4                                     <287>
5500 SYS DEC("2200"),,,3                            <H08>
5510 POKE A+2,0                                     <7O5>
5520 POKE A+3,TR                                    <UQ3>
5530 POKE A+4,SE                                    <RUJ>
5540 POKE A+5,1                                     <6SB>
5550 POKE DEC("00B0"),1                             <A6T>
5560 POKE DEC("00B1"),255                           <CR6>
5570 POKE DEC("00FB"),0                             <VS9>
5580 POKE DEC("00FC"),32                            <ID4>
5590 SYS DEC("2220"),,,6                            <OHE>
5600 DCLOSE                                         <9O7>
5610 GOSUB 1720                                     <VN1>
5620 SYS DEC("C98E")                                <5A6>
5630 GOTO 700                                       <32A>
```

Listing 1. Das Hauptprogramm »TOP-FLOP« (Schluß)

```
Name : top-flop.sub          2200 24f8
--------------------------------------
2200 : ad 1c 0a 29 bf 8d 1c 0a   50
2208 : a2 01 20 c9 ff a2 00 bd   fd
2210 : 30 14 20 d2 ff e8 88 d0   b8
2218 : f6 20 cc ff 2c 1c 0a 60   de
2220 : ad 1c 0a 29 bf 8d 1c 0a   70
2228 : a2 01 20 c9 ff a2 00 bd   1d
2230 : 30 14 20 d2 ff e8 88 d0   d8
2238 : f6 20 cc ff 2c 1c 0a 50   de
2240 : 4a 78 2c 0d dc a6 b0 ad   94
2248 : 00 dd 49 10 8d 00 dd a9   2f
2250 : 08 2c 0d dc f0 fb ad 00   f3
2258 : dd 49 10 8d 00 dd ad 0c   4d
2260 : dc 85 fa 29 0f c9 02 b0   8b
2268 : 22 a0 00 a9 08 2c 0d dc   df
2270 : f0 fb ad 00 dd 49 10 8d   4d
2278 : 00 dd ad 0c dc 91 fb c8   30
2280 : ea ea d0 e7 e6 fc ca d0   33
2288 : e2 18 24 38 58 60 ad 1c   fe
2290 : 0a 29 bf 8d 1c 0a a2 01   6f
2298 : 20 c9 ff a2 00 bd 30 14   c8
22a0 : 20 d2 ff e8 88 d0 f6 20   71
22a8 : cc ff 2c 1c 0a 70 01 60   eb
22b0 : 78 a9 40 85 fd a6 b0 a0   d7
22b8 : 00 ad 05 d5 09 08 8d 05   9c
22c0 : d5 a9 7f 8d 0d dc a9 00   5a
22c8 : 8d 05 dc a9 03 8d 04 dc   ab
22d0 : ad 0e dc 29 80 09 55 8d   a1
22d8 : 0e dc 2c 0d dc ad 00 dd   f8
22e0 : cd 00 dd d0 f8 45 fd 29   43
22e8 : 40 f0 f2 a5 fd 49 40 85   48
22f0 : fd b1 fb 8d 0c dc a9 08   d5
22f8 : 2c 0d dc f0 fb c8 ea ea   87
2300 : d0 db e6 fc ca d0 d6 a9   f9
2308 : 08 8d 0e dc ad 05 d5 29   a3
2310 : f7 8d 05 d5 2c 0d dc ad   c4
2318 : 00 dd 49 10 8d 00 dd a9   ff
2320 : 08 2c 0d dc ea ad 0c dc   23
2328 : 85 fa ad 00 dd 29 ef 8d   98
2330 : 00 dd a5 fa 29 0f c9 02   1e
2338 : 58 60 a9 30 85 fb a9 02   14
2340 : 85 fc a9 04 aa a0 00 20   1e
2348 : ba ff a9 00 20 bd ff 20   9c
2350 : c0 ff a2 04 20 c9 ff a5   d5
2358 : fc a2 12 20 cc cd a5 fb   f8
2360 : a2 13 20 cc cd a2 1f 20   dc
2368 : da cd 29 7f aa 18 69 40   f5
2370 : c9 80 b0 06 c9 60 90 02   4c
2378 : e9 40 e0 60 b0 06 e0 40   05
2380 : 90 02 e9 20 20 d2 ff b0   8a
2388 : 10 e6 fb d0 02 e6 fc a5   bb
2390 : fb c9 2f a5 fc e9 07 90   4d
2398 : be a9 0d 20 d2 ff a9 04   4e
23a0 : 20 c3 ff 20 cc ff 60 a2   39
23a8 : 00 a9 0f 8d ff 21 a0 10   9e
23b0 : bd 00 20 e8 20 a5 b8 88   b6
23b8 : f0 0d c0 08 d0 f2 20 7d   80
23c0 : ff 20 20 00 4c b0 23 ad   0a
23c8 : ff 21 f0 11 20 7d ff 0d   be
23d0 : 1d 1d 1d 1d 1d 1d 00 ce   bf
23d8 : ff 21 4c ae 23 60 a2 00   10
23e0 : a0 00 b1 84 18 c9 5b 90   dc
23e8 : 48 a5 c0 f0 07 b1 84 18   91
23f0 : c9 7b 90 48 a9 2e 20 0c   49
23f8 : c0 18 c0 07 d0 05 a9 20   f1
2400 : 20 0c c0 c8 18 c0 10 d0   d9
2408 : d9 e8 e0 10 f0 39 a5 84   08
2410 : 69 10 85 84 a9 00 85 f4   0e
2418 : 20 7d ff 0a 9d 9d 9d 9d   b0
2420 : 9d 9d 9d 9d 9d 9d 9d 9d   1f
2428 : 9d 9d 9d 9d 9d 00 4c e0   7c
2430 : 23 18 c9 20 90 b3 20 0c   15
2438 : c0 4c fa 23 18 c9 61 90   b8
2440 : b3 20 0c c0 4c fa 23 60   08
2448 : a2 00 a0 00 b1 84 18 c9   46
2450 : 1b 90 64 18 c9 40 90 48   41
2458 : a5 c0 f0 07 b1 84 18 c9   ae
2460 : 5b 90 48 a9 2e 20 0c c0   e0
2468 : 18 c0 07 d0 05 a9 20 20   1b
2470 : 0c c0 c8 18 c0 10 d0 d4   8b
2478 : e8 e0 10 f0 39 a5 84 69   98
2480 : 10 85 84 a9 00 85 f4 20   e9
2488 : 7d ff 0a 9d 9d 9d 9d 9d   b4
2490 : 9d 9d 9d 9d 9d 9d 9d 9d   8f
2498 : 9d 9d 9d 9d 00 4c 4a 24   f3
24a0 : 18 c9 20 90 b3 20 0c c0   a5
24a8 : 4c 69 24 18 c9 40 90 b3   fd
24b0 : 20 0c c0 4c 69 24 60 69   9c
24b8 : 40 20 0c c0 4c 69 24 8e   e1
24c0 : c5 24 a0 00 b9 00 20 99   0f
24c8 : 00 0e c8 c0 40 d0 f5 60   3d
24d0 : a2 05 20 c6 ff a0 00 20   1b
24d8 : cf ff 99 00 20 c8 d0 f7   89
24e0 : 20 cc ff 60 a2 05 20 c9   d9
24e8 : ff a0 00 b9 00 20 20 d2   96
24f0 : ff c8 d0 f7 20 cc ff 60   af
```

Listing 2. »TOP-FLOP.SUB« Die Maschinen-Unterroutinen von Top-Flop. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben.

Den illegalen Tracks auf der Spur

Der »Disc-Scanner 40« ist kein Disketten-Monitor im herkömmlichen Sinne. Mit ihm lassen sich vielmehr bislang unzugängliche »Orte« der Diskette bearbeiten: die »illegalen« Spuren 36 bis 40.

Schon seit einiger Zeit nutzten professionelle Programme die Möglichkeit, die Speicherkapazität der Diskette durch Nutzung der Tracks 36 bis 40 zu erhöhen. Dieses Verfahren ist recht einfach zu handhaben. Da es aber gerade für den Einsteiger sicher nicht leicht sein wird, die Floppy 1541 auf 40 Tracks umzuprogrammieren, wäre ein komfortabler Editor für den Zugriff auf die Spuren 36 bis 40 wünschenswert.

Genau in diese Lücke springt das Programm »Disc-Scanner 40«. Dieses Programm soll nicht als reiner Disk-Monitor dienen, sondern ein komfortables Hilfsprogramm zum Experimentieren mit den sogenannten illegalen Tracks sein.

Durch die komfortable Bedienung und die sichere Handhabung (Abfangen von Disk-Errors, Device-Test und ähnliches) läßt sich der »Disc-Scanner 40« von jedermann problemlos nutzen.

Bevor Sie jedoch mit dem Disc-Scanner auf die Jagd nach illegalen Spuren gehen, müssen Sie zunächst das Programm in Listing 1 abtippen und auf Diskette speichern. Verwenden Sie dazu bitte den MSE, zu dem Sie Erläuterungen auf Seite 158 dieser Ausgabe finden.

Abschließend kann der Disc-Scanner mit

```
LOAD "DISC-SCANNER 40",8
```

geladen und mit RUN gestartet werden.

Die Funktionen des »Disc-Scanner 40«

Nach dem Starten erscheint eine Bildschirmmaske, wie man sie in Bild 1 sehen kann. Sie ist in vier Teile gegliedert:
1. Teil: Auswahlmenü
2. Teil: Status-Zeile
3. Teil: Ein- und Ausgabefeld
4. Teil: Zeile für Meldungen

Im Auswahlmenü werden alle Funktionen und ihr Aufruf (Zifferntasten) dargestellt. Durch Drücken der entsprechenden Nummer gelangt man in die jeweilige Funktion.

In der Status-Zeile wird der Fehlerkanal des Diskettenlaufwerks dargestellt. Wird ein Funktionsaufruf erwartet, erscheint hier »PLEASE CHOOSE«.

Im Ein- und Ausgabefeld werden alle Ein- und Ausgaben getätigt. Hier erscheint beispielsweise der editierbare Block einer Diskette.

In der Meldungszeile gibt das Programm Aufforderungen wie »PLEASE SWITCH ON DISC DRIVE« oder Informationen über die jeweilige Funktion, wie etwa die Nummer des aktuellen Tracks und Sektors. Beschäftigen wir uns nun mit den Funktionen im Auswahlmenü:

1. READ BLOCK

Hat man durch Drücken der Zifferntaste <1> die Funktion »READ BLOCK« gewählt, kann man zunächst entscheiden, ob ein neuer Block von Diskette gelesen oder nur der Inhalt des Datenpuffers angezeigt werden soll.

Will man einen Block lesen, wird getestet, ob das Laufwerk bereit ist. Ist das Laufwerk nicht angeschaltet, wird dies vom Programm erkannt, und eine entsprechende Meldung ausgegeben. Liegt keine Diskette im Laufwerk, so gibt das Programm ebenfalls eine Meldung aus und wartet, bis eine Diskette eingelegt wurde. Durch <F1> wird diese Funktion übersprungen. Treten Schreib- oder Lesefehler auf, führt die Floppy keinen sogenannten »BUMP« aus. Auf diese Weise wird die Laufwerkmechanik bei Disk-Errors durch das lautstarke Rattern des Schreib-/Lesekopfes nicht strapaziert.

Jetzt muß der gewünschte Track und Sektor eingegeben werden. Die Input-Routine des C 64 wurde so umgeschrie-

```
DISC-SCANNER 40 (C) 1987 MARC HAWLITZECK
 (1) READ BLOCK           (2) WRITE BLOCK
 (3) SEND DISC COMMAND (4) EXIT
STATUS: 00, OK,00,00
> 00: 12 04 82 11 00 52 45 43   .....REC
> 08: 48 45 4E 50 52 4F 2E 20   HENPRO.
> 10: 43 20 36 34 20 00 00 00   C 64 ...
> 18: 00 00 00 00 00 00 34 00   ......4.
> 20: 00 00 82 13 00 41 42 53   .....ABS
> 28: 45 4E 44 45 52 3A A0 A0   ENDER:..
> 30: A0 A0 A0 A0 A0 00 00 00   ........
> 38: 00 00 00 00 00 00 01 00   ........
> 40: 00 00 82 13 01 48 2E 4A   .....H.J
> 48: 2E 20 56 4F 4C 4C 4D 45   . VOLLME
> 50: 52 A0 A0 A0 A0 00 00 00   R.......
> 58: 00 00 00 00 00 00 01 00   ........
> 60: 00 00 82 13 02 41 4B 54   .....AKT

LAST TRACK AND SEKTOR : 12 01
```

Bild 1. Der übersichtliche Bildschirmaufbau von »Disc-Scanner 40« sorgt für eine problemlose Bedienung

ben, daß ein Zerstören der Bildschirmmaske durch Umherfahren mit dem Cursor oder durch Steuerzeichen unmöglich gemacht wird. Wurde ein Block fehlerfrei gelesen, wird er auf dem Bildschirm dargestellt.

Zuerst erscheinen nur die ersten 8 Byte des Blockes. Durch Drücken der SHIFT-Taste werden weitere 8 Byte und ihr ASCII-Dump dargestellt. Ist das Ein- und Ausgabefeld voll, scrollt sein Inhalt um eine Zeile nach oben. Durch Drücken der RUN/STOP-Taste gelangt man während des Listvorganges in den Editiermodus. Nach den letzten 8 Byte wird automatisch in diesen Modus gesprungen. Mit den Cursortasten steuert man den Cursor. Das gerade angesteuerte Zeichen erscheint dabei revers. Mit dem Cursor kann man das Feld, in dem der Hex-Dump dargestellt wird, nicht verlassen. Nun lassen sich Änderungen im Hex-Dump vornehmen, wie es Bild 2 zeigt. Die Änderungen werden durch Drücken der RETURN-Taste übernommen. Es wird kein Carriage Return ausgeführt, dafür blinkt die übernommene Zeile zweimal hell auf. Der ASCII-Dump wird augenblicklich angepaßt. Durch Drücken von <F7> gelangt man wieder in die List-Routine. Durch Betätigen von <F1> kehrt man ins Auswahlmenü zurück.

Während der Bearbeitung eines Blockes wird in der Meldungszeile die Track- und Sektornummer angezeigt.

Gelegentlich kann es geschehen, daß nach einem Disk-Error keine Fehlermeldung erscheint, während die Floppystation weiterläuft. In einem solchen Fall öffnen Sie kurz das Laufwerk und schließen es sofort wieder. Die jeweilige Fehlermeldung erscheint dann auf dem Bildschirm und es kann normal weitergearbeitet werden.

2. WRITE BLOCK

Bei dieser Funktion muß man zunächst die Nummer des Tracks und Sektors eingeben, in der der Pufferinhalt geschrieben werden soll.

Zur besseren Orientierung werden die Track- und Sektornummer des zuvor gelesenen Blockes angegeben.

Nach ordnungsgemäßer Durchführung befindet man sich schließlich wieder im Auswahlmenü.

3. SEND DISC COMMAND

Hier können Sie entweder normale Befehle an das Laufwerk senden, oder eine komfortable Formatier-Routine anwählen. Trifft die Wahl auf den ersten Punkt, können Sie nun das gewünschte Kommando eingeben. Nach Ausführung des Kommandos erscheint die jeweilige Meldung der

```
DISC-SCANNER 40 (C) 1987 MARC HAWLITZECK
 (1) READ BLOCK            (2) WRITE BLOCK
 (3) SEND DISC COMMAND (4) EXIT
STATUS: 00, OK,00,00
> 00:  4B 01 01 01 01 01 01 01   K.......
> 08:  20 53 50 55 52 20 01 01    SPUR ..
> 10:  01 33 39 20 57 49 52 44   .39 WIRD
> 18:  47 45 52 41 44 45 01 01   GERADE..
> 20:  45 44 49 54 49 45 52 54   EDITIERT
> 28:  01 01 01 01 01 01 01 01   ........
> 30:  01 01 01 01 01 01 01 01   ........
> 38:  01 01 01 01 01 01 01 01   ........
> 40:  01 01 01 01 01 01 01 01   ........
> 48:  01 01 01 01 01 01 01 01   ........
> 50:  01 01 01 01 01 01 01 01   ........
> 58:  01 01 01 01 01 01 01 01   ........
> 60:  01 01 01 01 01 01 01 01   ........
> 68:  01 01 01 01 01 01 01 01   ........
LAST TRACK AND SEKTOR : 27 02
```

Bild 2. »Disk-Scanner 40« in Aktion: Die Hex-Werte der Diskettenblöcke können schnell mit dem Cursor verändert werden.

Floppystation. Danach wird wieder in das Auswahlmenü zurückgesprungen.

Bei der zweiten Option (FORMAT) handelt es sich um eine schnelle Formatier-Routine, bei der man zwischen drei Arten wählen kann:

1. Formatieren von Track 1 bis $23 ($23=35 dez.; alle Angaben hexadezimal)
2. Formatieren von Track 1 bis $29 (41 dez.)
3. Formatieren von Track x bis y

Die erste Funktion entspricht der normalen Formatier-Routine. Sie ist aber wesentlich schneller als die des Original Commodore-DOS.

Die zweite Funktion formatiert die gesamte Diskette, einschließlich der »illegalen« Tracks 36 bis 40.

Mit der dritten Funktion lassen sich schließlich beliebige Tracks formatieren. Hat man sich für eine der drei Funktionen entschieden, kann man anschließend wählen, ob ein »BUMP« (Justage des Schreib-/Lesekopfs durch Anschlagen am Chassis) ausgeführt werden soll. Wurden vor dem Formatier-Vorgang alle Schreib- und Lesekommandos fehlerfrei durchgeführt, ist ein »BUMP« nicht nötig. Haben Sie aber auf die Tracks 36 bis 40 zurückgegriffen, oder erschien eine Fehlermeldung, so sollten Sie mit »BUMP« formatieren.

Nun wird der Diskettenname eingegeben. Bei der dritten Funktion drücken Sie hier einfach RETURN. Jetzt ist die Disk einzugeben. Achten Sie bitte darauf, daß bei Wahl der dritten Form die ID der neu formatierten Tracks mit der der übrigen Tracks übereinstimmt, da Sie sonst einen »DISK ID MISMATCH«-Error erhalten. Bei den ersten beiden Funktionen startet anschließend die Formatierung und ist nach etwa 35 Sekunden beendet.

Bei der dritten Funktion müssen noch Start- und Zieltrack eingegeben werden. Es wird inklusive dem eingegebenen Zieltrack formatiert. Natürlich wird hierbei das Directory nicht neu erstellt.

Die Formatier-Routine stammt größtenteils aus dem Floppy-Kurs von Karsten Schramm, der ebenfalls in diesem Sonderheft zu finden ist.

EXIT

Mit dieser Funktion verlassen Sie den »Disc-Scanner 40«. Die Rahmen-, Vordergrund- und Zeichenfarbe, die während des Programmablaufs verändert wurde, werden wieder in den ursprünglichen Zustand zurückversetzt.

Hat man den »Disc-Scanner 40« verlassen, kann man ihn mit »SYS 35000« von neuem starten.

Was man noch wissen muß...

Alle Ein- und Ausgaben bei »Disc-Scanner 40« müssen hexadezimal vorgenommen werden.

Der Begriff »Pufferinhalt« wurde mehrmals erwähnt. Mit dieser Bezeichnung sind nicht etwa die Puffer in der Floppystation gemeint, sondern ein Bereich innerhalb des Programms, in dem die Blocks von Diskette abgelegt werden.

Nach dem Ende jeder Funktion muß eine Taste gedrückt werden, bevor das Programm ins Auswahlmenü zurückkehrt. Dadurch werden etwaige Fehlermeldungen nicht durch die Ausgabe von »PLEASE CHOOSE« gelöscht.

Manchmal kann es vorkommen, daß bereits nach einem Leseversuch ein »READ ERROR« erscheint. In diesem Fall hätte die Floppy normalerweise einen »BUMP« ausgeführt. Da diese Routine aber unterdrückt wird, kann sich der Schreib-/Lesekopf nicht ordnungsgemäß positionieren und die geforderten Daten lesen oder schreiben.

Dieser Trick, der das Rattern des Schreib-/Lesekopfes unterdrückt, läßt sich auch in Basic einfach realisieren:

```
OPEN1,8,15 : PRINT** # 1, "M-W" CHR$(106) CHR$(0)
CHR$(1) CHR$(193) : CLOSE1
```

Doch nun wünschen wir Ihnen viel Spaß, wenn Sie mit »Disc-Scanner 40« nach illegalen Spuren und Sektoren fahnden.

(Marc Hawlitzeck/Michael Thomas)

```
Name : discscanner 40      0801 1b45
-------------------------------------
0801 : 0b 08 c1 07 9e 32 30 36   0a
0809 : 31 00 00 00 a9 2c a0 08   c9
0811 : 85 5f 84 60 a9 45 a0 1b   f0
0819 : 85 5a 84 5b a9 d1 a0 9b   3b
0821 : 85 58 84 59 20 bf a3 4c   46
0829 : b8 88 00 a9 00 8d 21 d0   ed
0831 : a9 04 8d 20 d0 a9 80 8d   bb
0839 : 8a 02 a9 db a0 88 20 1e   b5
0841 : ab a9 a0 8d e7 07 a9 05   02
0849 : 8d e7 db 4c b9 89 93 12   a5
0851 : 9e 44 49 53 43 2d 53 43   40
0859 : 41 4e 4e 45 52 20 34 30   55
0861 : 20 28 43 29 20 31 39 38   6c
0869 : 37 20 4d 41 52 43 20 48   7c
0871 : 41 57 4c 49 54 5a 45 43   4e
0879 : 4b 92 1e 11 20 28 31 29   11
0881 : 20 52 45 41 44 20 42 4c   2b
0889 : 4f 43 4b 20 20 20 20 20   15
0891 : 20 20 20 28 32 29 20 57   6a
0899 : 52 49 54 45 20 42 4c 4f   31
08a1 : 43 4b 8e 08 0d 11 20 28   59
08a9 : 33 29 20 53 45 4e 44 20   fb
08b1 : 44 49 53 43 20 43 4f 4d   cb
08b9 : 4d 41 4e 44 20 28 34 29   29
08c1 : 20 45 58 49 54 0d 0d 12   c9
08c9 : 53 54 41 54 55 53 3a 20   3a
08d1 : 20 20 20 20 20 20 20 20   d1
08d9 : 20 20 20 20 20 20 20 20   d9
08e1 : 20 20 20 20 20 20 20 20   e1
08e9 : 20 20 20 20 20 20 20 20   e9
08f1 : 92 11 11 11 11 11 11 11   72
08f9 : 11 11 11 11 11 11 11 11   f9
0901 : 11 11 12 20 20 20 20 20   e7
0909 : 20 20 20 20 20 20 20 20   09
0911 : 20 20 20 20 20 20 20 20   11
0919 : 20 20 20 20 20 20 20 20   19
0921 : 20 20 20 20 20 20 20 20   21
0929 : 20 20 92 00 4c 04 8a 20   4d
0931 : e4 ff f0 03 4c d5 8a ae   ac
0939 : 3c 03 e8 8e 3c 03 f0 0a   b6
0941 : a2 00 e8 e0 64 d0 fb 4c   8f
0949 : bc 89 a2 00 8e 3c 03 ad   a5
0951 : 3d 03 d0 08 a2 01 8e 3d   2c
0959 : 03 4c f1 89 a2 00 8e 3d   0f
```

Listing 1. Der »Disc-Scanner 40« als Listing im MSE-Format. Bitte mit dem MSE (Seite 158) eingeben.

```
0961 : 03 4c 04 8a a2 00 bd 1a  32
0969 : 8a 18 69 40 9d f8 04 e8  e5
0971 : e0 0d d0 f2 4c bc 89 a2  80
0979 : 00 8e 3d 03 bd 27 8a 18  e0
0981 : 69 80 9d f8 04 e8 e0 0d  d6
0989 : d0 f2 4c bc 89 50 4c 45  54
0991 : 41 53 45 a0 43 48 4f 4f  34
0999 : 53 45 3f 3f 3f 3f 3f 3f  b0
09a1 : 20 3f 3f 3f 3f 3f 3f 13  2a
09a9 : 11 11 11 11 11 11 11 11  a9
09b1 : 11 11 11 11 11 11 11 43  16
09b9 : 4f 4d 4d 41 4e 44 20 3a  26
09c1 : 20 00 13 11 11 11 11 11  c8
09c9 : 11 1d 1d 1d 1d 1d 1d 1d  bd
09d1 : 1d 00 93 53 59 53 33 35  a5
09d9 : 30 30 30 20 28 52 45 54  04
09e1 : 55 52 4e 20 54 4f 20 27  86
09e9 : 44 49 53 43 2d 53 43 41  0c
09f1 : 4e 4e 45 52 20 34 30 27  b5
09f9 : 29 0d 54 48 41 4e 4b 53  21
0a01 : 20 46 4f 52 20 55 53 49  ef
0a09 : 4e 47 20 54 48 45 20 27  0b
0a11 : 44 49 53 43 2d 53 43 41  34
0a19 : 4e 4e 45 52 27 20 21 09  34
0a21 : 00 44 49 53 43 20 43 4f  e1
0a29 : 4d 4d 41 4e 44 20 4f 52  5e
0a31 : 20 53 50 45 45 44 20 46  3b
0a39 : 4f 52 4d 41 54 20 28 46  a0
0a41 : 31 2f 46 33 29 20 3f 00  92
0a49 : a8 98 c9 33 f0 15 c9 34  5d
0a51 : f0 79 c9 31 f0 07 c9 32  69
0a59 : f0 06 4c bc 89 4c 00 90  13
0a61 : 4c d0 96 20 7f 8b 20 74  7d
0a69 : 8b a9 c5 a0 96 20 1e ab  88
0a71 : a9 ae a0 8a 20 1e ab 20  cd
0a79 : e4 ff f0 fb c9 85 f0 07  b3
0a81 : c9 86 d0 f3 4c 0f 98 20  20
0a89 : 74 8b 20 a2 8c a9 34 a0  47
0a91 : 8a 20 1e ab a9 1c 20 99  58
0a99 : 8b a9 01 a2 08 a0 0f 20  8f
0aa1 : ba ff a9 00 20 bd ff 20  f5
0aa9 : c0 ff a2 01 20 c9 ff a2  c7
0ab1 : 00 bd 2a 8c 20 d2 ff e8  16
0ab9 : ec 52 8c d0 f4 20 cc ff  8f
0ac1 : a9 01 20 c3 ff 20 53 8c  d3
0ac9 : 4c c0 8d a9 0e 8d 20 d0  7d
0ad1 : 8d 86 02 a9 06 8d 21 d0  4a
0ad9 : a9 00 8d 8a 02 a9 5f a0  63
0ae1 : 8a 20 1e ab 4c 74 a4 a2  b9
0ae9 : 07 20 ff e9 e8 e0 18 d0  d5
0af1 : f8 60 a2 00 a9 e0 9d f8  cc
0af9 : 04 e8 e0 20 d0 f6 60 a2  39
0b01 : 00 a9 e0 9d c0 07 e8 e0  6b
0b09 : 28 d0 f6 60 8d 20 8c a2  b4
0b11 : 00 86 02 a9 2e 9d 2a 8c  9c
0b19 : e8 e0 28 d0 f8 a9 a4 20  45
0b21 : d2 ff 20 e4 ff f0 fb a6  5c
0b29 : cb e0 01 f0 53 4c 1d 8c  e8
0b31 : c9 20 f0 1a c9 14 d0 07  18
0b39 : ae 2a 8c e0 2e f0 e3 c9  c9
0b41 : 2a b0 07 c9 14 f0 07 4c  3c
0b49 : af 8b c9 5b 10 d4 a4 02  da
0b51 : 99 2a 8c 85 fa a9 14 20  61
0b59 : d2 ff b9 2a 8c 20 d2 ff  f4
0b61 : a9 a4 20 d2 ff a5 fa c9  6b
0b69 : 14 d0 10 a9 2e a4 02 99  62
0b71 : 2a 8c 88 c6 02 99 2a 8c  8b
0b79 : 4c af 8b e6 02 4c af 8b  b5
0b81 : a9 9d 20 d2 ff a9 20 20  69
0b89 : d2 ff a4 02 8c 52 8c 60  13
0b91 : a6 02 e0 17 d0 9a c9 14  84
0b99 : d0 88 4c bd 8b 2e 2e 2e  b7
0ba1 : 2e 2e 2e 2e 2e 2e 2e 2e  a1
0ba9 : 2e 2e 2e 2e 2e 2e 2e 2e  a9
0bb1 : 2e 2e 2e 2e 2e 2e 2e 2e  b1
0bb9 : 2e 2e 2e 2e 2e 2e 2e 2e  b9
0bc1 : 2e 2e 2e 2e 2e 00 a9 12  05
0bc9 : 20 d2 ff a9 4f a0 8a 20  ec
0bd1 : 1e ab a9 00 85 90 a9 01  b5
0bd9 : a2 08 a0 6f 20 ba ff a9  c0
0be1 : 00 20 bd ff 20 c0 ff a9  bc
0be9 : 08 20 b4 ff a9 6f 20 96  f2
0bf1 : ff 20 a5 ff 8d a1 8c 20  c2
0bf9 : d2 ff 20 a5 ff 20 d2 ff  d4
0c01 : 24 90 50 f6 a9 08 20 ab  13
0c09 : ff a9 01 20 c3 ff a9 92  29
0c11 : 20 d2 ff 60 00 20 66 8d  5c
0c19 : 20 8c 8b a9 01 a2 08 a0  1e
0c21 : 0f 20 ba ff a9 00 20 bd  85
0c29 : ff 20 c0 ff a2 01 20 c9  af
0c31 : ff a2 00 bd 14 8d 20 d2  0d
0c39 : ff e8 e0 07 d0 f5 20 cc  9c
0c41 : ff a2 01 20 c9 ff a9 49  ab
0c49 : 20 d2 ff 20 cc ff 20 7f  22
0c51 : 8b a9 01 20 c3 ff 20 53  59
0c59 : 8c ad a1 8c c9 30 d0 04  1f
0c61 : 20 8c 8b 60 20 9d 8d a9  2f
0c69 : 1b a0 8d 20 1e ab 20 e4  c5
0c71 : ff f0 fb c9 85 d0 07 20  5c
0c79 : 8c 8b 20 7f 8b 60 20 74  e8
0c81 : 8b 20 7f 8b 4c a2 8c 4d  14
0c89 : 2d 57 6a 00 01 c1 50 4c  f4
0c91 : 45 41 53 45 20 49 4e 53  21
0c99 : 45 52 54 20 44 49 53 43  83
0ca1 : 20 5b 52 45 54 55 52 4e  82
0ca9 : 5d 20 4f 52 20 5b 46 31  8d
0cb1 : 5d 92 00 50 4c 45 41 53  fc
0cb9 : 45 20 53 57 49 54 43 48  a3
0cc1 : 20 4f 4e 20 44 49 53 43  83
0cc9 : 2d 44 52 49 56 45 20 5b  9d
0cd1 : 52 45 54 55 52 4e 5d 92  b8
0cd9 : 00 a9 01 a2 08 a0 0f 20  44
0ce1 : ba ff a9 01 a2 9c a0 8d  d2
0ce9 : 20 bd ff 20 c0 ff b0 05  c5
0cf1 : a9 01 4c c3 ff 20 9d 8d  39
0cf9 : a9 40 a0 8d 20 1e ab 20  7c
0d01 : e4 ff f0 fb a9 01 20 c3  4b
0d09 : ff 20 8c 8b 4c 66 8d 20  1b
0d11 : a9 a5 a0 8d 20 1e ab 60  c9
0d19 : 13 11 11 11 11 11 11 11  1b
0d21 : 11 11 11 11 11 11 11 11  21
0d29 : 11 11 11 11 11 11 11 11  29
0d31 : 11 12 00 20 9d 8d a9 26  89
0d39 : a0 8e 20 1e ab 20 e4 ff  3b
0d41 : f0 fb 20 74 8b 20 7f 8b  94
0d49 : 20 8c 8b 4c bc 89 48 4a  e9
0d51 : 4a 4a 4a 20 f4 8d aa 68  8e
0d59 : 29 0f 20 f4 8d 48 8a 20  36
0d61 : d2 ff 68 20 d2 ff 60 18  30
0d69 : 69 f6 90 02 69 06 69 3a  93
0d71 : 60 a9 19 a0 8e 20 1e ab  ba
0d79 : 38 ad 3f 03 e9 08 18 20  38
0d81 : db 8d a9 3a 20 d2 ff a9  c0
0d89 : 20 4c d2 ff 3e 20 00 13  8f
0d91 : 11 11 11 11 11 11 11 11  91
0d99 : 00 50 52 45 53 53 20 41  d1
0da1 : 4e 59 20 4b 45 59 20 54  56
0da9 : 4f 20 52 45 54 55 52 4e  1b
0db1 : 20 54 4f 20 49 4e 50 55  c6
0db9 : 54 2d 44 45 53 43 92 00  f7
0dc1 : 52 45 41 44 20 42 4c 4f  73
0dc9 : 43 4b 20 4f 52 20 53 45  a2
0dd1 : 45 20 42 55 46 46 45 52  b2
0dd9 : 20 28 46 31 2f 46 33 29  09
0de1 : 20 3f 00 2e 2e 2e 2e 2e  d0
0de9 : 2e 2e 2e 2e 2e 2e 2e 2e  e9
0df1 : 2e 2e 2e 2e 2e 2e 2e 2e  f1
0df9 : 2e 2e 2e 2e 2e 2e 2e 2e  f9
0e01 : 2e 2e 2e 2e 2e 2e 2e 2e  01
0e09 : 2e 2e 2e 2e 2e 2e 2e 2e  09
0e11 : 2e 2e 2e 2e 2e 2e 2e 2e  11
0e19 : 2e 2e 2e 2e 2e 2e 2e 2e  19
0e21 : 2e 2e 2e 2e 2e 2e 2e 2e  21
0e29 : 2e 2e 2e 2e 2e 2e 2e 2e  29
0e31 : 2e 2e 2e 2e 2e 2e 2e 2e  31
0e39 : 2e 2e 2e 2e 2e 2e 2e 2e  39
0e41 : 2e 2e 2e 2e 2e 2e 2e 2e  41
0e49 : 2e 2e 2e 2e 2e 2e 2e 2e  49
0e51 : 2e 2e 2e 2e 2e 2e 2e 2e  51
0e59 : 2e 2e 2e 2e 2e 2e 2e 2e  59
0e61 : 2e 2e 2e 2e 2e 2e 2e 2e  61
0e69 : 2e 2e 2e 2e 2e 2e 2e 2e  69
0e71 : 2e 2e 2e 2e 2e 2e 2e 2e  71
0e79 : 2e 2e 2e 2e 2e 2e 2e 2e  79
0e81 : 2e 2e 2e 2e 2e 2e 2e 2e  81
0e89 : 2e 2e 2e 2e 2e 2e 2e 2e  89
0e91 : 2e 2e 2e 2e 2e 2e 2e 2e  91
0e99 : 2e 2e 2e 2e 2e 2e 2e 2e  99
0ea1 : 2e 2e 2e 2e 2e 2e 2e 2e  a1
0ea9 : 2e 2e 2e 2e 2e 2e 2e 2e  a9
0eb1 : 2e 2e 2e 2e 2e 2e 2e 2e  b1
0eb9 : 2e 2e 2e 2e 2e 2e 2e 2e  b9
0ec1 : 2e 2e 2e 2e 2e 2e 2e 2e  c1
0ec9 : 2e 2e 2e 2e 2e 2e 2e 2e  c9
0ed1 : 2e 2e 2e 2e 2e 2e 2e 2e  d1
0ed9 : 2e 2e 2e 2e 2e 2e 2e 2e  d9
0ee1 : 2e 2e 2e 2c 2c 2c 2c 2c  65
0ee9 : 2c 2c 2c 2c 2c 2c 2c 2c  e9
0ef1 : 2c 2c 2c 2c 2c 2c 2c 2c  f1
0ef9 : 2c 2c 2c a9 06 85 0a a9  84
0f01 : 06 85 0b a2 02 86 f9 a9  71
0f09 : 80 05 7f a6 f9 8d 4d 02  06
0f11 : ad 4d 02 20 20 03 4c 99  68
0f19 : d5 ea ea 48 86 f9 8a 0a  9d
0f21 : aa b5 07 8d 4d 02 b5 06  e1
0f29 : f0 2d cd 40 eb b0 28 aa  65
0f31 : 68 48 29 f0 c9 90 d0 2f  e8
0f39 : 68 48 4a b0 05 ad 01 01  32
0f41 : 90 03 ad 02 01 f0 05 cd  46
0f49 : d5 fe d0 13 8a 20 4b f2  f1
0f51 : cd 4d 02 f0 02 b0 10 20  8a
0f59 : 52 d5 a9 66 4c 45 e6 20  98
0f61 : 52 d5 a9 73 4c 45 e6 a6  4e
0f69 : f9 68 8d 4d 02 95 00 9d  ab
0f71 : 5b 02 60 a9 c5 a0 96 20  17
0f79 : 1e ab a9 4d a0 8e 20 1e  bc
0f81 : ab 20 e4 ff f0 fb c9 85  96
0f89 : f0 07 c9 86 d0 f3 4c 73  05
0f91 : 92 20 74 8b 20 27 90 4c  d8
0f99 : 7d 91 a9 01 20 c3 ff 20  ca
0fa1 : a2 8c 20 9d 8d a9 a5 a0  43
0fa9 : 90 20 1e ab a9 80 a0 90  89
0fb1 : 20 1e ab a9 30 a2 47 8d  51
0fb9 : cd 8b 8e d8 8b a9 02 20  59
0fc1 : 99 8b ad 2a 8c 8d ce 90  62
0fc9 : ad 2b 8c 8d cf 90 a9 98  3a
0fd1 : a0 90 20 1e ab a9.02 20  d6
0fd9 : 99 8b a9 23 a2 5b 8d cd  dd
0fe1 : 8b 8e d8 8b ad 2a 8c 8d  d4
0fe9 : d0 90 ad 2b 8c 8d d1 90  70
0ff1 : 4c d2 90 13 92 11 11 11  45
0ff9 : 11 11 11 11 11 11 11 11  f9
1001 : 11 54 52 41 43 4b 20 20  48
1009 : 3a 20 00 0d 0d 0d 53 45  06
1011 : 4b 54 4f 52 20 3a 20 00  f9
1019 : 49 4e 50 55 54 53 20 49  3b
1021 : 4e 20 48 45 58 2d 43 4f  d5
1029 : 44 45 20 41 4e 44 20 57  76
1031 : 49 54 48 20 54 57 4f 20  38
1039 : 4e 55 4d 42 45 52 53 92  27
1041 : 00 00 00 00 00 ad ce 90  0b
1049 : c9 3a 90 09 20 01 91 8d  e0
1051 : ce 90 4c e8 90 38 e9 30  6a
1059 : 8d ce 90 ad cf 90 c9 3a  44
1061 : 90 09 20 01 91 8d cf 90  84
1069 : 4c 09 91 38 e9 30 8d cf  9b
1071 : 90 4c 09 91 18 69 01 29  bf
1079 : 07 09 08 60 ad ce 90 20  e7
1081 : 25 91 0a 0a 0a 0a 8d ce  f8
1089 : 90 ad cf 90 20 25 91 0d  81
1091 : ce 90 8d ce 90 4c 2e 91  2c
1099 : c9 3a 90 02 69 08 29 0f  7d
10a1 : 60 ad d0 90 c9 3a 90 09  e1
10a9 : 20 01 91 8d d0 90 4c 44  ab
10b1 : 91 38 e9 30 8d d0 90 ad  dc
10b9 : d1 90 c9 3a 90 09 20 01  60
10c1 : 91 8d d1 90 4c 5a 91 38  ee
10c9 : e9 30 8d d1 90 ad d0 90  43
10d1 : 20 74 91 0a 0a 0a 0a 8d  05
10d9 : d0 90 ad d1 90 20 74 91  96
10e1 : 0d d0 90 8d d0 90 60 c9  d3
10e9 : 3a 90 02 69 08 29 0f 60  e0
10f1 : 20 83 91 4c 2d 92 20 74  92
10f9 : 8b 20 8c 8b ad ce 90 ae  1a
1101 : d0 90 8d 89 8f 8e 8d 8f  71
1109 : a9 88 a2 8f 85 fa 86 fb  d3
1111 : a9 00 a2 03 85 fc 86 fd  1a
1119 : a9 08 20 b1 ff a9 6f 20  50
1121 : 93 ff a9 4d 20 a8 ff a9  62
1129 : 2d 20 a8 ff a9 57 20 a8  b7
1131 : ff a0 00 a5 fc 20 a8 ff  a8
1139 : a5 fd 20 a8 ff a9 1e 20  00
1141 : a8 ff b1 fa 20 a8 ff c8  8d
1149 : c0 1e d0 f6 20 ae ff 18  d3
1151 : a5 fa 69 1e 85 fa a5 fb  50
1159 : 69 00 85 fb 18 a5 fc 69  19
1161 : 1e 85 fc a5 fd 69 00 85  6c
1169 : fd 18 a5 fc c9 78 d0 a8  70
1171 : a9 08 20 b1 ff a9 6f 20  a8
1179 : 93 ff a9 4d 20 a8 ff a9  ba
1181 : 2d 20 a8 ff a9 45 20 a8  7f
1189 : ff a9 00 20 a8 ff a9 03  98
1191 : 20 a8 ff a9 08 20 ae ff  77
1199 : a9 01 20 c3 ff 4c 53 8c  0c
11a1 : ad a1 8c c9 30 f0 03 4c  aa
11a9 : c0 8d a9 0d a8 a2 08 20  3c
11b1 : ba ff a9 02 a2 71 a0 92  73
11b9 : 20 bd ff 20 c0 ff 20 81  4b
11c1 : 93 20 93 93 a2 0d 20 c6  5c
11c9 : ff a2 00 20 cf ff 9d 70  72
11d1 : 8e e8 d0 f7 20 cc ff a9  c2
11d9 : 0d 20 c3 ff a9 0f 20 c3  02
11e1 : ff 4c 73 92 23 32 20 74  63
11e9 : 8b 20 8c 8b a9 9d a0 8d  3e
11f1 : 8d db 96 8c dc 96 20 d6  54
11f9 : 96 a9 1e a0 ab 8d db 96  c3
1201 : 8c dc 96 a9 1c a0 8e 20  17
1209 : 1e ab a0 00 8c 3e 03 8c  05
1211 : 40 03 8c 8d 02 a0 08 8c  06
1219 : 3f 03 20 fe 8d ac 3e 03  ff
1221 : b9 70 8e 20 db 8d a9 20  cb
1229 : 20 d2 ff ac 3e 03 c8 ee  45
1231 : 3e 03 cc 3f 03 d0 e9 20  aa
1239 : 43 93 ad 40 03 c9 0e b0  d1
1241 : 30 ee 40 03 a9 0d 20 d2  82
1249 : ff 18 ad 3f 03 69 08 8d  5e
1251 : 3f 03 18 20 e4 ff c9 88  a2
1259 : f0 8c c9 85 d0 03 4c cf  a8
```

```
1261 : 8d ad 8d 02 c9 01 f0 b2  36
1269 : a5 91 c9 7f d0 e5 4c f6  94
1271 : 93 c9 1f f0 3e ee 40 03  31
1279 : 18 ad 3f 03 69 08 8d 3f  24
1281 : 03 18 20 e4 ff c9 88 d0  47
1289 : 03 4c 73 92 c9 85 d0 03  f3
1291 : 4c cf 8d ad 8d 02 c9 01  f0
1299 : f0 09 a5 91 c9 7f d0 e2  4b
12a1 : 4c f6 93 20 b8 93 a9 91  43
12a9 : 20 d2 ff a9 0d 20 d2 ff  84
12b1 : 4c a7 92 4c f6 93 a9 20  f2
12b9 : 20 d2 ff 38 ad 3f 03 e9  fe
12c1 : 08 18 aa bd 70 8e c9 20  1a
12c9 : b0 04 a9 2e d0 18 c9 5f  5f
12d1 : 90 11 c9 c0 b0 04 a9 2e  a2
12d9 : d0 0c c9 db 90 05 a9 2e  d1
12e1 : 4c 73 93 bd 70 8e 20 d2  25
12e9 : ff a9 00 85 d4 e8 ec 3f  34
12f1 : 03 d0 d0 60 a9 0f a2 08  4a
12f9 : a0 0f 20 ba ff a9 00 20  0e
1301 : bd ff 20 c0 ff 60 a9 08  98
1309 : 20 b1 ff a9 6f 20 93 ff  7d
1311 : a0 00 b9 b0 93 20 a8 ff  13
1319 : c8 c0 08 d0 f5 a9 08 20  6a
1321 : ae ff 60 42 2d 50 20 31  67
1329 : 33 20 30 a9 40 a2 05 85  e6
1331 : fa 86 fb a9 68 a2 05 85  5d
1339 : fc 86 fd a0 00 b1 fc 91  b0
1341 : fa c8 d0 f9 e6 fb e6 fd  f8
1349 : a0 00 b1 fc 91 fa c8 d0  ab
1351 : f9 e6 fb e6 fd a0 00 b1  e1
1359 : fc 91 fa c8 c0 30 d0 f7  b6
1361 : 60 a0 00 ad 43 03 91 fa  50
1369 : 60 a9 46 a2 05 85 fa 86  f9
1371 : fb a9 01 8d 42 03 a9 04  1e
1379 : 8d 47 03 a9 06 8d 41 03  78
1381 : a0 00 b1 fa 8d 43 03 20  2c
1389 : ee 93 a0 00 b1 fa 09 80  81
1391 : 91 fa 20 e4 ff f0 fb c9  4f
1399 : 1d f0 15 c9 9d f0 5a c9  0b
13a1 : 91 d0 03 4c d2 94 c9 11  00
13a9 : d0 03 4c f2 94 4c 28 95  e4
13b1 : ad 41 03 4a b0 20 ad 41  4e
13b9 : 03 c9 15 b0 cd 20 ee 93  bd
13c1 : ee 41 03 ee 47 03 18 a5  27
13c9 : fa 69 01 85 fa a5 fb 69  08
13d1 : 00 85 fb 4c 0d 94 ad 41  cb
13d9 : 03 c9 15 b0 ad ee 41 03  7a
13e1 : ee 47 03 ee 47 03 20 ee  fc
13e9 : 93 18 a5 fa 69 02 85 fa  04
13f1 : a5 fb 69 00 85 fb 4c 0d  72
13f9 : 94 ad 41 03 4a 90 20 ad  1a
1401 : 41 03 c9 06 90 84 20 ee  82
1409 : 93 ce 41 03 ce 47 03 38  58
1411 : a5 fa e9 01 85 fa a5 fb  8d
1419 : e9 00 85 fb 4c 0d 94 ad  be
1421 : 41 03 c9 07 b0 03 4c 17  ba
1429 : 94 ce 41 03 ce 47 03 ce  a6
1431 : 47 03 20 ee 93 38 a5 fa  67
1439 : e9 02 85 fa a5 fb e9 00  c6
1441 : 85 fb 4c 0d 94 ad 42 03  3e
1449 : c9 02 b0 03 4c 17 94 ce  0d
1451 : 42 03 20 ee 93 38 a5 fa  82
1459 : e9 28 85 fa a5 fb e9 00  f9
1461 : 85 fb 4c 0d 94 38 ad 3f  d9
1469 : 03 e9 10 18 4a 4a 4a cd  24
1471 : 42 03 b0 03 4c 17 94 20  d1
1479 : 1d 95 ee 42 03 20 ee 93  79
1481 : 18 a5 fa 69 28 85 fa a5  3d
1489 : fb 69 00 85 fb 4c 0d 94  69
1491 : ad 42 03 c9 0f b0 01 60  95
1499 : 4c 17 94 c9 85 d0 03 4c  53
14a1 : cf 8d c9 88 d0 03 4c 73  f7
14a9 : 92 c9 0d f0 60 aa ad 41  16
14b1 : 03 c9 06 b0 03 4c 1f 94  69
14b9 : 8a c9 30 b0 03 4c 1f 94  82
14c1 : c9 41 b0 07 c9 3a 90 03  ef
14c9 : 4c 1f 94 c9 47 90 03 4c  a1
14d1 : 1f 94 c9 41 b0 0a a0 00  b3
14d9 : 91 fa 8d 43 03 4c 3d 94  64
14e1 : 38 e9 40 18 a0 00 91 fa  67
14e9 : 8d 43 03 4c 3d 94 20 8f  7a
14f1 : 95 0a 0a 0a 0a 8d 44 03  73
14f9 : ad 45 03 20 8f 95 0d 44  70
1501 : 03 60 c9 3a 08 29 0f 28  44
1509 : 90 02 69 08 60 38 a5 fa  4a
1511 : ed 47 03 85 fa a5 fb e9  b4
1519 : 00 85 fb 18 a0 01 b1 fa  ad
1521 : 8d 45 03 a0 00 b1 fa c9  33
1529 : 30 b0 03 18 69 40 20 7b  85
1531 : 95 8d 46 03 a0 04 a2 00  34
1539 : b1 fa 29 7f c9 30 b0 03  88
1541 : 18 69 40 9d 70 8f 8a 4a  14
1549 : b0 09 c8 e8 e0 10 d0 e8  71
1551 : 4c e7 95 c8 c8 e8 e0 10  87
1559 : d0 de a0 01 a2 00 b9 70  d3
1561 : 8f 8d 45 03 88 b9 70 8f  a0
1569 : 20 7b 95 9d 80 8f c8 c8  99
1571 : c8 e8 e0 08 d0 e8 a2 00  c5
1579 : ac 46 03 bd 80 8f 99 70  8d
1581 : 8e e8 c8 e0 08 d0 f4 18  dd
1589 : ad 42 03 69 07 85 d6 a9  91
1591 : 1f 85 d3 20 6c e5 ae 46  a9
1599 : 03 a0 00 bd 70 8e c9 20  87
15a1 : b0 04 a9 2e d0 18 c9 5f  37
15a9 : 90 11 c9 c0 b0 04 a9 2e  7a
15b1 : d0 0c c9 db 90 05 a9 2e  a9
15b9 : 4c 4b 96 bd 70 8e 20 d2  aa
15c1 : ff a9 00 85 d4 e8 c8 c0  7f
15c9 : 08 d0 d0 20 6c 96 18 a5  99
15d1 : fa 6d 47 03 85 fa a5 fb  73
15d9 : 69 00 85 fb 4c 1f 94 18  63
15e1 : a5 fa 69 fe 85 fc a5 fb  0c
15e9 : 69 d3 85 fd 20 89 96 a2  4b
15f1 : 3c a0 00 c8 d0 fd ca d0  60
15f9 : f8 4c 89 96 a9 01 a0 00  72
1601 : 91 fc c8 c0 27 d0 f9 a2  81
1609 : 00 a0 00 c8 d0 fd e8 e0  d5
1611 : 1e d0 f6 a9 05 a0 00 91  03
1619 : fc c8 c0 27 d0 f9 60 4c  85
1621 : 41 53 54 20 54 52 41 43  88
1629 : 4b 20 41 4e 44 20 53 45  bc
1631 : 4b 54 4f 52 20 3a 20 00  19
1639 : 13 11 11 11 11 11 11 11  3b
1641 : 11 11 00 20 d6 96 4c fa  28
1649 : 96 a9 c5 a0 96 20 1e ab  73
1651 : a9 ac a0 96 20 1e ab ad  48
1659 : ce 90 20 db 8d a9 20 20  da
1661 : d2 ff ad d0 90 20 db 8d  4d
1669 : a9 92 4c d2 ff 20 27 90  87
1671 : a9 0d a8 a2 08 20 ba ff  8c
1679 : a9 02 a2 71 a0 92 20 bd  95
1681 : ff 20 c0 ff 20 81 93 20  5d
1689 : 93 93 a2 0d 20 c9 ff a2  c6
1691 : 00 bd 70 8e 20 d2 ff e8  c8
1699 : d0 f7 20 cc ff a9 0d 20  c8
16a1 : c3 ff a9 0f 20 c3 ff a9  24
16a9 : 90 8d 95 8f 20 83 91 a9  0f
16b1 : 80 8d 95 8f 4c c0 8d 31  b2
16b9 : 2e 20 53 50 45 45 44 20  a6
16c1 : 46 4f 52 4d 41 54 20 54  cd
16c9 : 52 41 43 4b 20 31 20 2d  5d
16d1 : 20 32 33 0d 0d 32 2e 20  d4
16d9 : 53 50 45 45 44 20 46 4f  4b
16e1 : 52 4d 41 54 20 54 52 41  25
16e9 : 43 4b 20 31 20 2d 20 32  50
16f1 : 39 0d 0d 33 2e 20 53 50  2c
16f9 : 45 45 44 20 46 4f 52 4d  b9
1701 : 41 54 20 54 52 41 43 4b  d2
1709 : 20 58 20 2d 20 59 00 44  58
1711 : 49 53 43 4e 41 4d 45 20  72
1719 : 20 20 3a 20 00 0d 44 49  e8
1721 : 53 43 49 44 20 20 20 20  b5
1729 : 20 3a 20 00 13 92 11 11  9b
1731 : 11 11 11 11 11 11 11 11  31
1739 : 11 11 11 53 54 41 52 54  c3
1741 : 54 52 41 43 4b 20 3a 20  56
1749 : 00 0d 0d 45 4e 44 54 52  b9
1751 : 41 43 4b 20 20 20 3a 20  37
1759 : 00 46 4f 52 4d 41 54 20  0b
1761 : 57 49 54 48 20 42 55 4d  7f
1769 : 50 20 28 59 2f 4e 29 20  49
1771 : 3f 00 50 4c 45 41 53 45  84
1779 : 20 57 41 49 54 20 2e 2e  1a
1781 : 2e 00 20 74 8b a9 c5 a0  a4
1789 : 96 20 1e ab a9 44 a0 97  9b
1791 : 20 1e ab 20 e4 ff f0 fb  b9
1799 : 8d 44 03 c9 31 90 f4 c9  41
17a1 : 34 b0 f0 20 74 8b a9 c5  43
17a9 : a0 96 20 1e ab a9 e6 a0  45
17b1 : 97 20 1e ab 20 e4 ff f0  60
17b9 : fb c9 59 f0 0c c9 4e d0  f7
17c1 : f3 a9 00 8d 6a 9b 4c 5b  a6
17c9 : 98 a9 c0 8d 6a 9b 20 66  e8
17d1 : 8d 20 53 8c 20 7f 8b 20  41
17d9 : 74 8b a9 c5 a0 96 20 1e  b1
17e1 : ab a9 9c a0 97 20 1e ab  e6
17e9 : a9 10 20 99 8b a0 00 b9  07
17f1 : 2a 8c 99 b3 9b c8 c4 02  55
17f9 : d0 f5 a9 2c 99 b3 9b c8  eb
1801 : 8c b1 9b a9 0d 20 d2 ff  9f
1809 : a9 aa a0 97 20 1e ab a9  17
1811 : 02 20 99 8b ac b1 9b ad  1d
1819 : 2a 8c 99 b3 9b c8 ad 2b  73
1821 : 8c 99 b3 9b c8 8c b2 9b  cd
1829 : ad 44 03 c9 31 f0 07 c9  3d
1831 : 32 f0 10 4c dd 98 a9 01  b4
1839 : 8d af 9b a9 24 8d b0 9b  62
1841 : 4c 21 99 a9 01 8d af 9b  2c
1849 : a9 2a 8d b0 9b 4c 21 99  55
1851 : a9 b9 a0 97 8d 3a 90 8c  f8
1859 : 3c 90 a9 d6 a0 97 8d 5c  d8
1861 : 90 8c 5e 90 20 2f 90 a9  f2
1869 : 80 a0 90 8d 3a 90 8c 3c  e2
1871 : 90 a9 98 a0 90 8d 5c 90  18
1879 : 8c 5e 90 ad ce 90 8d af  15
1881 : 9b ae d0 90 e8 8e b0 9b  b6
1889 : a9 ea a2 00 9d 88 9b e8  ae
1891 : e0 03 d0 f8 a9 d3 a2 99  3d
1899 : 85 fa 86 fb a9 00 a2 05  ec
18a1 : 85 fc 86 fd a9 08 20 b1  c4
18a9 : ff a9 6f 20 93 ff a9 4d  d7
18b1 : 20 a8 ff a9 2d 20 a8 ff  d1
18b9 : a9 57 20 a8 ff a0 00 a5  7b
18c1 : fc 20 a8 ff a5 fd 20 a8  13
18c9 : ff a9 1e 20 a8 ff b1 fa  70
18d1 : 20 a8 ff c8 c0 1e 90 f6  8b
18d9 : 20 ae ff 18 a5 fa 69 1e  67
18e1 : 85 fa 90 03 e6 fb 18 a5  62
18e9 : fc a6 fd 69 1e 85 fc 90  08
18f1 : 02 e6 fd e0 07 90 ad c9  41
18f9 : 00 90 a9 20 8c 8b 20 9d  91
1901 : 8d a9 ff a0 97 20 1e ab  c1
1909 : a9 08 20 b1 ff a9 6f 20  40
1911 : 93 ff a9 4d 20 a8 ff a9  52
1919 : 2d 20 a8 ff a9 45 20 a8  17
1921 : ff a9 60 20 a8 ff a9 06  4e
1929 : 20 a8 ff 20 ae ff 20 53  b3
1931 : 8c 20 8c 8b a9 20 a2 40  08
1939 : a0 ee 8d 88 9b 8e 89 9b  50
1941 : 8c 8a 9b 4c c0 8d ea a5  f2
1949 : 0a c9 24 90 07 a9 12 85  64
1951 : 43 4c 13 05 20 4b f2 85  53
1959 : 43 a9 00 85 1b a0 00 a2  1e
1961 : 01 a5 39 99 00 03 c8 c8  83
1969 : a5 1b 99 00 03 c8 a5 0a  23
1971 : 99 00 03 c8 a5 13 99 00  3e
1979 : 03 c8 a5 12 99 00 03 c8  c3
1981 : a9 0f 99 00 03 c8 99 00  f5
1989 : 03 c8 a9 00 59 fa 02 59  83
1991 : fb 02 59 fc 02 59 fd 02  6a
1999 : 99 f9 02 e6 1b a5 1b c5  63
19a1 : 43 90 be a9 03 85 31 98  63
19a9 : 48 8a 9d 00 07 e8 d0 fa  8f
19b1 : 20 30 fe 68 a8 88 20 e5  d1
19b9 : fd 20 f5 fd a9 07 85 31  4f
19c1 : 20 e9 f5 85 3a 20 8f f7  d7
19c9 : a9 00 85 32 20 0e fe a9  dc
19d1 : ff 8d 01 1c a2 05 50 fe  ec
19d9 : b8 ca d0 fa a2 0a a4 32  fb
19e1 : 50 fe b8 b9 00 03 8d 01  66
19e9 : 1c c8 ca d0 f3 a2 09 50  4f
19f1 : fe b8 a9 55 8d 01 1c ca  47
19f9 : d0 f5 a9 ff a2 05 50 fe  c0
1a01 : b8 8d 01 1c ca d0 f7 a2  9c
1a09 : bb 50 fe b8 bd 00 01 8d  be
1a11 : 01 1c e8 d0 f4 a0 00 50  69
1a19 : fe b8 b1 30 8d 01 1c c8  c9
1a21 : d0 f5 a9 55 a2 08 50 fe  aa
1a29 : b8 8d 01 1c ca d0 f7 a5  ca
1a31 : 32 18 69 0a 85 32 c6 1b  46
1a39 : d0 95 50 fe b8 50 fe b8  43
1a41 : 20 00 fe a9 c8 85 1f a9  df
1a49 : 00 85 30 a9 03 85 31 a5  ba
1a51 : 43 85 1b 20 56 f5 a2 0a  d5
1a59 : a0 00 50 fe b8 ad 01 1c  22
1a61 : d1 30 d0 0e c8 ca d0 f2  4c
1a69 : 18 a5 30 69 0a 85 30 4c  b3
1a71 : 35 06 c6 1f d0 d1 a9 06  8d
1a79 : 4c d3 fd 20 56 f5 a0 bb  41
1a81 : 50 fe b8 ad 01 1c d9 00  8c
1a89 : 01 d0 e7 c8 d0 f2 a2 fc  2e
1a91 : 50 fe b8 ad 01 1c d9 00  9c
1a99 : 07 d0 d7 c8 ca d0 f1 c6  a0
1aa1 : 1b d0 b0 4c 9e fd a0 00  36
1aa9 : b9 e0 06 99 00 02 c8 cc  54
1ab1 : df 06 90 f4 ad df 06 8d  63
1ab9 : 74 02 ad de 06 8d 7b 02  34
1ac1 : a9 00 85 7f 20 00 c1 ac  1e
1ac9 : 7b 02 b9 00 02 85 12 b9  bc
1ad1 : 01 02 85 13 20 07 d3 a9  74
1ad9 : 1a ea ea ea a9 00 85 00  31
1ae1 : a5 00 30 fc ae dc 06 86  29
1ae9 : 0a a9 e0 85 02 a5 02 30  66
1af1 : fc c9 02 b0 0c e8 ec dd  e0
1af9 : 06 90 ec 20 40 ee 60 ea  59
1b01 : ea a2 02 4c 0a e6 01 01  24
1b09 : 01 01 01 01 01 01 01 01  09
1b11 : 01 01 01 01 01 01 01 01  11
1b19 : 01 01 01 01 01 01 01 01  19
1b21 : 01 01 00 00 00 00 01 01  a9
1b29 : 01 01 01 01 01 01 01 01  29
1b31 : 01 01 01 01 01 01 01 01  31
1b39 : 00 ff ff 00 00 ff 00 00  39
1b41 : 00 ff 00 98 e9 f5 85 3a  2d
```

Listing 1. »Disc-Scanner 40« (Schluß)

Der Sieger der Backup-Renner

»Master-Copy parallel« (MCP) ist das schnellste Kopierprogramm für die Floppy 1541. Für ein Back-Up einer kompletten Disketten-Seite benötigt das Programm nur 18 Sekunden.

Man kommt kaum mit dem Wechseln der Disketten nach, so schnell arbeitet »Master-Copy parallel«. Es kopiert 40 Tracks, Lesefehler werden erkannt und auf der Zieldiskette repariert. Diese Spitzenwerte erreicht das Kopierprogramm natürlich nur mit einem Parallelkabel zwischen dem C 64 und der 1541. Für alle Leser, die ein solches Kabel noch nicht besitzen, haben wir die Verbindungsliste des Parallelkabels vom User-Port des C 64 zur Floppystation (CIA 6522) in Tabelle 1 aufgeführt. Die benötigten Teile sind in einer Stückliste angegeben (Tabelle 2).

»Master-Copy parallel« repariert Disketten beim Kopiervorgang. Es arbeitet also nicht wie die sogenannten Nibbler, die Fehler auf die Kopie übertragen. Die Fehler, die von »MCP« erkannt werden, sind im einzelnen:

1. READ ERRORS 20,21,22,23,27,29;
2. zerstörte und unformatierte Tracks;
3. »Killertracks«, bei denen kein Header existiert.

Damit ist das Duplikat auf jeden Fall von allen Fehlern befreit. Treten beim Kopieren Lesefehler auf, erhöht sich die benötigte Zeit allerdings geringfügig. Das Programm bietet außerdem die Möglichkeit, in nur 9 Sekunden alle 749 Blocks einer Diskette mit 40 Spuren zu überprüfen.

Nach dem Start des Programms erscheint das Auswahlmenü (Bild 1). Von diesem Menü rufen Sie alle Funktionen per Tastendruck auf. Folgende Befehle stehen zur Verfügung:

<A>: In einem Kopierdurchgang können ein bis zehn Duplikate erstellt werden. Dadurch sparen Sie auch bei Massenkopien erheblich Zeit.

<B>: Floppy-Befehle, zum Beispiel zum Formatieren einer Diskette, können an das Laufwerk gesendet werden.

<D>: Das Inhaltsverzeichnis der Diskette (Directory) wird durch Anwahl dieser Funktion am Bildschirm aufgelistet. Jeweils 20 Einträge stellt das Programm auf dem Bildschirm dar. Nach dem Drücken einer Taste werden weitere 20 Einträge gezeigt. Mit einer beliebigen Taste gelangt man anschließend zum Hauptmenü.

<K>: Startet den Kopiervorgang. Zunächst wird geprüft, ob die eingelegte Diskette initialisierbar ist und ob ein Parallelkabel den C 64 und das Diskettenlaufwerk verbindet. Tritt bei diesen Abfragen ein Fehler auf, zeigt »Master-Copy parallel« diesen in der unteren Bildschirmzeile an. Ansonsten wird auf einen zweiten Bildschirm umgeschaltet, auf dem die Quell-Diskette und die Ziel-Diskette symbolisiert sind. Das Programm erwartet die jeweils invertiert dargestellte Diskette im Laufwerk. Zwischen beiden Symbolen befindet sich das Info-Feld, in dem der Durchgang (PASS), die Fehleranzahl, die benötigte Kopierzeit sowie die Nummer der Ziel-Diskette (wichtig für mehrere Kopien einer Diskette) dargestellt sind. Ein Lesefehler wird durch die Meldung »READ ERROR IN XX« angezeigt. Ist ein Track zerstört oder nicht formatiert, wird ein »SYNC NOT FOUND« (Track-Error) ausgegeben. Bei einem sogenannten Killertrack, bei dem kein Header gefunden wird, erscheint die Meldung »HEADER NOT FOUND«. Die reine Kopierzeit beträgt bei 40 Spuren nur 18 Sekunden. Alle Lesefehler repariert »Master-Copy parallel« auf der Ziel-Diskette.



Bild 1. Das Hauptmenü von »Master-Copy parallel«

<V>: Diese Funktion erlaubt, in nur 9 Sekunden alle 749 Blöcke einer Diskette auf Fehler zu untersuchen. Jeder Block ist am Bildschirm durch einen Punkt dargestellt. Die Sektoren eines Tracks werden in einem Raster von oben nach unten angezeigt. Die Tracks sind von links nach rechts aufgeführt. Geprüfte Tracks erkennt man an einem Stern über der Track-Nummer. Kann ein Block korrekt gelesen werden, bleibt der Punkt bestehen. Bei einem Lesefehler erscheint statt des Punktes ein »R«. Treten Track-Errors auf, das heißt eine ganze Spur kann nicht gelesen werden, wandelt das Programm diese Blockausgaben in ein »S« (für SYNC NOT FOUND) oder ein »H« (für HEAD NOT FOUND) um.

<S>: Hier wird die Nummer des ersten zu kopierenden Tracks eingegeben. Vorgegeben ist Track-Nummer 1.

<E>: Der letzte zu kopierende Track ist hier einzustellen. Vorgegeben ist Track 40. Die Nummer des End-Track muß größer als die Nummer des Start-Track sein.

<G>: Diese Funktion ändert die Adresse des angesprochenen Laufwerks. Erlaubt sind die Adressen 8 bis 11.

<P>: Beendet das Programm und löst einen Reset aus.

User-Port	CIA 6522
B	Pin 39
C	Pin 2
D	Pin 3
E	Pin 4
F	Pin 5
H	Pin 6
J	Pin 7
K	Pin 8
L	Pin 9
8	Pin 18

Tabelle 1. So wird ein Parallelkabel am User-Port des C 64 und an dem CIA 6522 in der Floppy angeschlossen

Stückliste	
1m	10adriges Flachbandkabel
1	User-Port-Stecker
2	IC-Sockel 40-polig (für erfahrene Löter reicht auch ein IC-Sockel. Es empfiehlt sich, Sockel mit gedrehten Beinchen zu kaufen.)

Tabelle 2. Die notwendigen Teile für den Anschluß des Parallelkabels

<I>: Gibt eine Kurzbeschreibung von »Master-Copy parallel« auf dem Bildschirm aus.

Eingabehinweis: Geben Sie »MASTER-COPY PAR.« (Listing 1) mit dem MSE ein. Beachten Sie bitte die Eingabehinweise auf Seite 158. Nach dem Speichern auf Diskette laden Sie »Master-Copy parallel« mit »LOAD "MASTER-COPY PAR.",8«. Starten Sie das Programm durch Eingabe von »RUN« und Drücken der Taste <RETURN>.

Mit Floppy-Beschleuniger wie Dolphin-DOS oder Prologic-DOS arbeitet »Master-Copy parallel« nicht zusammen. Das Programm erkennt das angeschlossene Parallelkabel nicht und meldet »kein Parallelkabel angeschlossen«. Benutzen Sie eine Speeddos-Version oder nur ein Parallelkabel, arbeitet »Master-Copy parallel« einwandfrei.

Schwierigkeiten treten auch auf, wenn Sie eine Floppy 1541c benutzen. Folgende Einsprünge der Floppy-ROM benutzt das Programm Routine:

fa2e	im Speicher ab 18bd (Schrittmotorsteuerung)
f6d0	im Speicher ab 192b und 193c (4 Binär- in 5 GCR-Byte wandeln)
fe00	im Speicher ab 1b95 (PCR auf Lesen umschalten)
fe24	im Speicher ab 1b1e (10240mal Schreiben)

Besonders der Einsprung in die Schrittmotor-Steuerung, der nicht über einen Zeiger im ROM erfolgen kann, führt zu den Problemen bei der Floppy 1541c. Sollte es Ihnen gelingen, das Programm für die 1541c anzupassen, dann würden wir uns über die Einsendung dieser Anpassung freuen.

(Frank Riemenschneider/rs)

```
Name : master-copy par.  0801 1c80
-----------------------------------
0801 : 35 08 0a 00 8f 14 14 14  cf
0809 : 14 14 14 2a 2a 2a 2a 2a  63
0811 : 2a 2a 2a 2a 2a 2a 2a 2a  11
0819 : 2a 2a 2a 2a 2a 2a 2a 2a  19
0821 : 2a 2a 2a 2a 2a 2a 2a 2a  21
0829 : 2a 2a 2a 2a 2a 2a 2a 2a  29
0831 : 2a 2a 2a 00 69 08 0b 00  fe
0839 : 8f 14 14 14 14 14 14 2a  e1
0841 : 4d 20 41 20 53 20 54 20  bb
0849 : 45 20 52 20 2d 20 43 20  58
0851 : 4f 20 50 20 59 20 20 50  80
0859 : 20 41 20 52 20 41 20 4c  91
0861 : 20 4c 20 45 20 4c 2a 00  65
0869 : 9d 08 0c 00 8f 14 14 14  20
0871 : 14 14 14 2a 2a 20 20 44  86
0879 : 45 52 20 57 45 4c 54 53  89
0881 : 43 48 4e 45 4c 4c 53 54  42
0889 : 45 20 31 35 34 31 2d 4b  e9
0891 : 4f 50 49 45 52 45 52 20  dc
0899 : 20 2a 2a 00 df 08 0d 00  cb
08a1 : 9e 32 34 35 33 3a 8f 14  78
08a9 : 14 14 14 14 14 14 14 14  a9
08b1 : 14 14 14 14 14 2a 2a 20  d2
08b9 : 28 43 29 20 31 39 38 37  fd
08c1 : 20 20 42 59 20 46 52 41  ad
08c9 : 4e 4b 20 52 49 45 4d 45  8e
08d1 : 4e 53 43 48 4e 45 49 44  5f
08d9 : 45 52 20 2a 2a 00 13 09  96
08e1 : 0e 00 8f 14 14 14 14 14  b0
08e9 : 14 2a 2a 2a 2a 2a 2a 2a  d3
08f1 : 2a 2a 2a 2a 2a 2a 2a 2a  f1
08f9 : 2a 2a 2a 2a 2a 2a 2a 2a  f9
0901 : 2a 2a 2a 2a 2a 2a 2a 2a  01
0909 : 2a 2a 2a 2a 2a 2a 2a 2a  09
0911 : 2a 00 00 00 a2 80 bd 20  a1
0919 : c2 9d 3c 05 ca 10 f7 a9  ba
0921 : 00 a2 08 85 69 86 6a a9  ed
0929 : 00 85 fa a5 fa 20 b3 cf  7e
0931 : e6 fa a5 fa c9 15 d0 f3  cd
0939 : a2 00 86 f9 a6 f9 bd 64  b6
0941 : 04 85 fa bd 6b 04 85 fb  63
0949 : a9 00 85 fc a9 08 85 fd  e0
0951 : a2 1b a0 00 b1 fc 91 fa  e8
0959 : c8 d0 f9 e6 fb e6 fd ca  69
0961 : d0 f0 e6 f9 a5 f9 c9 07  01
0969 : d0 d2 a2 80 bd 3c 05 9d  68
0971 : 20 c2 ca 10 f7 4c 0a c6  3f
0979 : 94 28 bc 50 e4 78 00 22  b1
0981 : 3d 57 72 8c a7 e0 a2 75  8f
0989 : bd 15 09 9d 00 04 ca 10  32
0991 : f7 4c a0 17 a9 9a a2 09  c5
0999 : 85 fa 86 fb a9 00 a2 c2  67
09a1 : 85 fc 86 fd a2 0e a0 00  23
09a9 : b1 fa 91 fc c8 d0 f9 e6  a4
09b1 : fb e6 fd ca d0 f0 4c 87  cd
09b9 : 09 20 fb ed a9 93 20 d2  ec
09c1 : ff a2 0a a0 00 4c 0c e5  06
09c9 : a9 77 8d 20 cc a9 c8 8d  ee
09d1 : 21 cc a9 8f 8d 27 cc a9  4d
09d9 : c8 8d 28 cc a9 85 8d f6  f6
09e1 : ca a9 8c 8d f7 ca a9 20  11
09e9 : 8d f8 ca a9 c1 8d 32 cc  c5
09f1 : a9 c8 8d 33 cc 4c 57 c9  e8
09f9 : 20 e9 c7 78 a9 35 85 01  6b
0a01 : a9 00 85 a7 85 a9 a9 fc  47
0a09 : a2 04 85 a8 86 aa a2 04  74
0a11 : a0 00 b1 a7 91 a9 c8 d0  3e
0a19 : f9 e6 a8 e6 aa ca d0 f0  b2
0a21 : a9 37 85 01 58 4c 1c ca  d5
0a29 : 20 e4 ff f0 fb c9 56 f0  22
0a31 : 0a c9 49 f0 03 4c 1e ce  39
0a39 : 4c 56 c4 a9 4c 8d f6 ca  b9
0a41 : a9 d5 8d f7 ca a9 c2 8d  57
0a49 : f8 ca a9 15 8d 32 cc a9  a4
0a51 : c4 8d 33 cc a9 2e 8d 27  d3
0a59 : cc a9 c4 8d 28 cc a9 2b  c2
0a61 : 8d 20 cc a9 c4 8d 21 cc  3d
0a69 : 20 57 c9 4c f3 cc 20 e9  2b
0a71 : c7 20 38 c3 78 a5 fa 85  7a
0a79 : fb a9 02 20 2c c8 a5 fb  65
0a81 : 20 2c c8 20 8c cb 20 b9  09
0a89 : cb e6 fb a5 fb c5 fe d0  06
0a91 : e8 a2 18 a0 01 20 0c e5  f2
0a99 : 20 28 c6 56 45 52 49 46  e2
0aa1 : 59 20 49 53 54 20 42 45  a1
0aa9 : 45 4e 44 45 54 2e 20 57  b5
0ab1 : 45 49 54 45 52 20 4d 49  47
0ab9 : 54 20 54 41 53 54 45 20  88
0ac1 : 21 00 58 20 fc c7 20 05  95
0ac9 : c8 a9 03 20 4e c4 c6 fe  4f
0ad1 : 60 20 28 c6 93 11 30 30  07
0ad9 : 30 30 30 30 30 30 30 31  db
0ae1 : 31 31 31 31 31 31 31 31  e1
0ae9 : 31 32 32 32 32 32 32 32  e8
0af1 : 32 32 32 33 33 33 33 33  2f
0af9 : 33 33 33 33 33 34 31 32  f7
0b01 : 33 34 35 36 37 38 39 30  dd
0b09 : 31 32 33 34 35 36 37 38  f9
0b11 : 39 30 31 32 33 34 35 36  0b
0b19 : 37 38 39 30 31 32 33 34  9b
0b21 : 35 36 37 38 39 30 00 4c  f4
0b29 : e3 c3 14 14 14 14 14 14  d0
0b31 : 14 14 14 14 14 14 14 14  31
0b39 : 14 14 14 12 12 12 12 12  bd
0b41 : 12 12 11 11 11 11 11 11  c3
0b49 : 10 10 10 10 10 10 10 10  49
0b51 : 10 10 78 a0 c8 f0 18 40  90
0b59 : 68 90 b8 e0 08 30 58 80  b8
0b61 : a8 d0 f8 20 48 70 98 04  26
0b69 : 04 04 04 05 05 05 05 05  a8
0b71 : 05 06 06 06 06 06 06 06  70
0b79 : 07 07 07 07 a9 28 85 a7  e8
0b81 : a6 a7 ca bd 91 c3 aa bd  c2
0b89 : b9 c3 85 a9 bd ce c3 85  27
0b91 : aa a4 a7 88 a9 2e 91 a9  2e
0b99 : ca 10 ec c6 a7 a5 a7 d0  67
0ba1 : df 20 f1 c5 ea 9d ff 03  67
0ba9 : ca e4 fa b0 f8 60 98 48  3f
0bb1 : b9 b9 c3 85 a9 b9 ce c3  13
0bb9 : 85 aa a4 fb 88 a9 12 91  7d
0bc1 : a9 4c db 03 a9 13 2c a9  1f
0bc9 : 08 48 a6 fb ca bd 91 c3  87
0bd1 : aa bd b9 c3 85 a9 bd ce  7b
0bd9 : c3 85 aa a4 fb 88 68 48  d4
0be1 : 91 a9 ca 10 ec 68 60 8d  aa
0be9 : 00 dd a9 00 4c 2c c8 a2  d1
0bf1 : 06 a0 03 20 0c e5 20 28  cd
0bf9 : c6 b0 c0 c0 c0 c0 c0 c0  f6
0c01 : c0 c0 c0 c0 c0 c0 c0 c0  00
0c09 : c0 c0 c0 c0 c0 c0 c0 c0  08
0c11 : c0 c0 c0 c0 c0 c0 c0 c0  10
0c19 : c0 c0 ae 00 a2 07 a0 03  d0
0c21 : 20 0c e5 20 28 c6 dd 20  35
0c29 : 2a 20 4d 41 53 54 45 52  70
0c31 : 2d 43 4f 50 59 20 50 41  38
0c39 : 52 41 4c 4c 45 4c 20 56  ac
0c41 : 20 31 2e 33 20 2a 20 dd  7b
0c49 : 00 a2 08 a0 03 20 0c e5  de
0c51 : 20 28 c6 dd 20 50 52 4f  5f
0c59 : 47 52 41 4d 4d 45 4e 54  a4
0c61 : 57 49 43 4b 4c 55 4e 47  ce
0c69 : 20 31 2f 38 37 2d 35 2f  05
0c71 : 38 37 3a 20 dd 00 a2 09  52
0c79 : a0 03 20 0c e5 20 28 c6  b2
0c81 : dd 20 20 20 20 20 46 52  3b
0c89 : 41 4e 4b 20 52 49 45 4d  e7
0c91 : 45 4e 53 43 48 4e 45 49  d9
0c99 : 44 45 52 20 20 20 20 20  dc
0ca1 : 20 dd 00 a2 0a a0 03 20  f6
0ca9 : 0c e5 20 28 c6 dd 20 47  1f
0cb1 : 45 4f 52 47 2d 42 55 45  e0
0cb9 : 43 48 4e 45 52 2d 53 54  e1
0cc1 : 52 2e 20 39 20 2d 20 33  ac
0cc9 : 20 48 20 37 31 20 dd 00  88
0cd1 : a2 0b a0 03 20 0c e5 20  bc
0cd9 : 28 c6 dd 20 4b 4f 50 49  e3
0ce1 : 45 52 5a 45 49 54 20 34  af
0ce9 : 30 20 54 52 41 43 4b 53  8b
0cf1 : 3a 20 31 38 20 53 45 4b  d7
0cf9 : 2e 20 20 dd 00 a2 0c a0  82
0d01 : 03 20 0c e5 20 28 c6 dd  ee
0d09 : 20 20 50 52 55 45 46 5a  e5
0d11 : 45 49 54 20 34 30 20 54  02
0d19 : 52 41 43 4b 53 3a 20 20  0e
0d21 : 39 20 53 45 4b 2e 20 20  cf
0d29 : dd 00 a2 0d a0 03 20 0c  0b
0d31 : e5 20 28 c6 dd 20 55 45  c8
0d39 : 42 45 52 54 52 41 47 55  34
0d41 : 4e 47 53 52 41 54 45 3a  92
0d49 : 20 32 37 2c 35 20 4b 42  dc
0d51 : 2f 53 45 4b 20 dd 00 a2  1b
0d59 : 0e a0 03 20 0c e5 20 28  3d
0d61 : c6 ad c0 c0 c0 c0 c0 c0  dd
0d69 : c0 c0 c0 c0 c0 c0 c0 c0  68
0d71 : c0 c0 c0 c0 c0 c0 c0 c0  70
0d79 : c0 c0 c0 c0 c0 c0 c0 c0  78
0d81 : c0 c0 bd 00 20 05 c8 4c  f7
0d89 : f3 cc a6 fc ca a9 2a 60  8f
0d91 : 3b 39 7f b7 ff 39 dd 76  6e
0d99 : bb 10 ff 00 ff bb 39 f2  05
0da1 : 39 ea ea a9 15 8d 18 d0  ff
0da9 : a9 01 85 fd 85 8f a9 00  6f
0db1 : 8d 20 d0 8d 21 d0 a9 08  83
0db9 : 85 f9 a9 28 85 fe 4c f3  14
0dc1 : cc 68 85 9b 68 85 9c a0  fc
0dc9 : 00 a9 1f d0 02 b1 9b e6  6a
0dd1 : 9b d0 02 e6 9c c9 00 f0  2c
0dd9 : 06 20 d2 ff 4c 34 c6 6c  fe
0de1 : 9b 00 f8 a2 08 85 9b a9  7d
0de9 : 00 85 9c 06 9b 65 9c 85  f6
0df1 : 9c ca d0 f7 d8 48 4a 4a  b3
0df9 : 4a 4a 20 65 c6 68 29 0f  90
0e01 : 18 69 30 4c d2 ff 48 a9  05
0e09 : 20 a0 00 99 00 04 88 d0  91
0e11 : fa a0 90 99 ff 04 88 d0  96
0e19 : fa 68 8d d5 c6 a5 d3 48  df
0e21 : a5 d6 48 8a 48 a2 01 a0  74
0e29 : 04 20 0c e5 20 bb c6 68  c9
0e31 : 8d d5 c6 a2 01 a0 1c 20  75
0e39 : 0c e5 20 bb c6 20 59 c7  1a
0e41 : 68 aa 68 a8 20 0c e5 a9  7b
0e49 : 63 a0 27 99 8f 05 88 d0  de
0e51 : fa 4c 1c ca ad d5 c6 8d  91
0e59 : eb c6 c9 12 d0 08 a9 92  75
0e61 : 8d e8 c6 4c d2 c6 a9 12  cc
0e69 : 8d e8 c6 20 28 c6 12 b0  82
0e71 : c0 c0 c0 c0 c0 c0 ae 9d  e1
0e79 : 9d 9d 9d 9d 9d 9d 9d 11  5f
0e81 : dd 92 20 20 12 20 20 d7  06
0e89 : 20 dd 9d 9d 9d 9d 9d 9d  2b
0e91 : 9d 9d 11 dd 20 20 d5 c9  eb
0e99 : 20 20 dd 9d 9d 9d 9d 9d  6d
0ea1 : 9d 9d 9d 11 dd 20 20 ca  8b
0ea9 : cb 20 20 dd 9d 9d 9d 9d  c1
0eb1 : 9d 9d 9d 9d 11 dd 20 20  f9
0eb9 : d5 c9 20 20 dd 9d 9d 9d  fb
0ec1 : 9d 9d 9d 9d 9d 11 dd 20  62
0ec9 : 20 dd dd 20 20 dd 9d 9d  f6
0ed1 : 9d 9d 9d 9d 9d 9d 11 dd  1f
0ed9 : 20 20 ca cb 20 20 dd 9d  eb
0ee1 : 9d 9d 9d 9d 9d 9d 9d 11  c7
0ee9 : ad c0 c0 c0 c0 c0 c0 bd  cf
0ef1 : 00 60 a2 01 a0 0e 20 0c  fd
0ef9 : e5 20 28 c6 92 50 41 53  29
```

Listing 1. »Master-Copy parallel« bitte mit dem MSE eingeben

```
0f01 : 53 3a 20 00 a5 61 20 49  f2
0f09 : c6 a2 03 a0 0e 20 0c e5  d3
0f11 : 20 28 c6 92 45 52 52 4f  18
0f19 : 52 53 3a 20 00 a5 62 20  9e
0f21 : 49 c6 a2 05 a0 0e 20 0c  2a
0f29 : e5 20 28 c6 5a 49 45 4c  9f
0f31 : 44 49 53 4b 3a 20 00 a6  4a
0f39 : 8c e8 8a 20 49 c6 a2 07  43
0f41 : a0 0e 20 0c e5 20 28 c6  ff
0f49 : 92 5a 45 49 54 3a 20 00  1a
0f51 : ad 0b dc ad 0a dc 20 dd  34
0f59 : c7 20 d2 ff a9 2e 20 d2  17
0f61 : ff ad 09 dc 48 20 e2 c7  b5
0f69 : 20 d2 ff 68 20 dd c7 20  50
0f71 : d2 ff ad 08 dc 60 29 0f  43
0f79 : 09 30 60 4a 4a 4a 4a 09  2e
0f81 : 30 60 ad 11 d0 29 ef 8d  a0
0f89 : 11 d0 a2 00 a0 96 e8 d0  af
0f91 : fd 88 d0 fa 60 ad 11 d0  bf
0f99 : 09 10 8d 11 d0 60 a9 00  e7
0fa1 : 85 c6 20 e4 ff f0 fb 60  66
0fa9 : a2 00 8e 03 dd a2 0b 8e  8c
0fb1 : 00 dd a2 03 2c 00 dd 10  03
0fb9 : fb ad 01 dd 8e 00 dd 2c  40
0fc1 : 00 dd 30 fb 60 a2 ff 8e  74
0fc9 : 03 dd a2 0b 8e 00 dd a2  6a
0fd1 : 03 2c 00 dd 10 fb 8d 01  bf
0fd9 : dd 8e 00 dd 2c 00 dd 30  54
0fe1 : fb 60 a9 00 8d 0b dc 60  dc
0fe9 : ad 08 dc 8d 08 dc 60 58  1d
0ff1 : 20 49 c8 e6 61 20 49 c9  95
0ff9 : 20 fc c7 20 05 c8 20 e9  78
1001 : c7 20 4f c8 78 60 58 20  f1
1009 : 49 c8 20 50 c9 4c 5f c8  d6
1011 : 20 a9 c8 20 28 c6 3a 53  84
1019 : 59 4e 43 20 4e 4f 54 20  5f
1021 : 46 4f 55 4e 44 00 78 60  15
1029 : 20 a9 c8 20 28 c6 3a 48  86
1031 : 45 41 44 45 52 20 4e 4f  cf
1039 : 54 20 46 4f 55 4e 44 00  f2
1041 : 78 60 20 28 c6 0d 54 52  c1
1049 : 41 43 4b 45 52 52 4f 52  41
1051 : 20 49 4e 20 00 a5 fb 4c  63
1059 : 49 c6 e6 62 98 48 20 b2  bd
1061 : cf 20 28 c6 0d 52 45 41  1e
1069 : 44 20 45 52 52 4f 52 2c  9a
1071 : 00 a5 fb 20 49 c6 a9 2c  10
1079 : 20 d2 ff 68 ea a8 20 49  16
1081 : c6 ea 78 60 a2 17 a0 01  4e
1089 : 20 0c e5 20 28 c6 44 49  89
1091 : 53 4b 45 54 54 45 20 49  e8
1099 : 53 54 20 4e 49 43 48 54  61
10a1 : 20 4c 45 53 42 41 52 20  5b
10a9 : 21 20 20 20 00 20 05 c8  8d
10b1 : 60 c6 fe 20 28 c6 0d 0d  3f
10b9 : 57 45 49 54 45 52 20 4d  92
10c1 : 49 54 20 54 41 53 54 45  52
10c9 : 4e 44 52 55 43 4b 00 58  b8
10d1 : 20 49 c8 20 49 c9 20 fc  29
10d9 : c7 20 05 c8 a9 03 8d 00  f4
10e1 : dd 60 a9 12 a2 92 4c 6d  66
10e9 : c6 a2 12 a9 92 4c 6d c6  89
10f1 : ad 03 dd 48 a9 ff 8d 03  77
10f9 : dd a9 6f 8d 01 dd a9 0f  fc
1101 : a6 f9 a8 20 ba ff a9 00  24
1109 : 20 bd ff 20 c0 ff a2 0f  c0
1111 : 20 c9 ff a9 49 20 d2 ff  2c
1119 : 20 cc ff a2 0f 20 c6 ff  01
1121 : 20 cf ff 48 20 cf ff c9  46
1129 : 0d d0 f9 20 cc ff 68 c9  23
1131 : 30 f0 04 68 4c eb c8 a2  74
1139 : 0f 20 c9 ff a2 04 bd f8  fe
1141 : c9 20 d2 ff ca 10 f7 20  1c
1149 : cc ff a2 0f 20 c6 ff 20  18
1151 : cf ff 48 20 cc ff a9 0f  c7
1159 : 20 c3 ff 68 aa 68 8d 03  92
1161 : dd e0 6f f0 31 a2 17 a0  6e
1169 : 01 20 0c e5 20 28 c6 4b  2f
1171 : 45 49 4e 20 50 41 52 41  cd
1179 : 4c 4c 45 4c 4b 41 42 45  19
1181 : 4c 20 56 4f 52 48 41 4e  66
1189 : 44 45 4e 20 21 00 4c 05  55
1191 : c8 18 01 52 2d 4d a6 fe  c6
1199 : e4 fd b0 06 a4 fd 84 fe  b3
11a1 : 86 fd e6 fe a9 93 20 d2  1d
11a9 : ff a2 0a a0 00 20 0c e5  8d
11b1 : 20 5f c2 f0 1d a9 06 a0  c8
11b9 : 00 99 00 d8 c8 d0 fa 99  d3
11c1 : 00 d9 c8 d0 fa 99 00 da  2c
11c9 : c8 d0 fa 99 00 db c8 d0  8f
11d1 : fa 60 a9 00 a2 04 85 a7  15
11d9 : 86 a8 a9 00 a2 03 85 a9  c9
11e1 : 86 aa a5 f9 20 0c ed a9  d2
11e9 : 6f 20 b9 ed a9 4d 20 dd  d6
11f1 : ed a9 2d 20 dd ed a9 57  a5
11f9 : 20 dd ed a0 00 a5 a9 20  ab
1201 : dd ed a5 aa 20 dd ed a9  8f
1209 : 20 20 dd ed b1 a7 20 dd  03
1211 : ed c8 c0 20 90 f6 20 fe  d6
1219 : ed 18 a5 a7 69 20 85 a7  6e
1221 : 90 03 e6 a8 18 a5 a9 a6  a4
1229 : aa 69 20 85 a9 90 03 e6  3a
1231 : aa e8 e0 07 90 ac a5 f9  61
1239 : 20 0c ed a9 6f 20 b9 ed  cb
1241 : a9 4d 20 dd ed a9 2d 20  76
1249 : dd ed a9 45 20 dd ed a9  2c
1251 : 0d 20 dd ed a9 03 20 dd  92
1259 : ed 20 20 c2 78 a9 00 85  97
1261 : 61 a9 00 85 62 a5 fd 85  9e
1269 : fa a5 fe 85 fc ad 0f dc  d9
1271 : 29 7f 8d 0f dc ad 0e dc  cc
1279 : 09 80 8d 0e dc a9 00 8d  1e
1281 : 0b dc 8d 0a dc 8d 09 dc  b7
1289 : 8d 08 dc a9 00 85 8b 85  ec
1291 : 8c 20 56 c8 a5 fa 85 fb  1c
1299 : a9 02 20 2c c8 a5 fb 20  bb
12a1 : 2c c8 20 8c cb 20 b9 cb  07
12a9 : e6 fb e6 8b a5 fb c5 fe  07
12b1 : f0 08 a6 8b e0 08 f0 02  d6
12b9 : d0 de a6 fa 86 8d 85 fa  e2
12c1 : a9 00 85 8c a6 8d 86 fb  46
12c9 : 20 6d c8 a9 00 85 8b a9  b5
12d1 : 03 20 2c c8 a5 fb 20 2c  1c
12d9 : c8 20 8c cb 20 7e cc e6  45
12e1 : fb e6 8b a5 fb c5 fa d0  62
12e9 : e6 85 8e e6 8c a5 8c c5  c6
12f1 : 8f d0 d1 a5 8e c5 fe f0  06
12f9 : 03 4c f2 ca 20 28 c6 0d  b1
1301 : 0d 0d 4b 4f 50 49 45 52  5b
1309 : 56 4f 52 47 41 4e 47 20  68
1311 : 49 53 54 20 42 45 45 4e  1d
1319 : 44 45 54 2e 00 a9 00 20  68
1321 : 2c c8 4c 18 c9 a6 8b e0  89
1329 : 07 d0 0a a9 00 85 69 a9  75
1331 : e0 85 6a d0 1c a9 00 85  a3
1339 : 69 a9 08 85 6a e0 00 f0  b9
1341 : 10 a5 69 18 69 94 85 69  a5
1349 : a5 6a 69 1a 85 6a ca d0  39
1351 : f0 60 20 3a cc a9 04 20  2b
1359 : 2c c8 a9 00 8d 03 dd 20  fc
1361 : 0f c8 aa 30 4f 0a aa bd  f0
1369 : 54 cc 18 65 69 85 a7 85  42
1371 : a9 bd 55 cc 65 6a 85 a8  f9
1379 : 69 01 85 aa a2 33 a0 00  60
1381 : 2c 00 dd 10 fb ad 01 dd  14
1389 : 91 a7 c8 2c 00 dd 30 fb  4d
1391 : ad 01 dd 91 a7 c8 d0 e8  3e
1399 : 2c 00 dd 10 fb ad 01 dd  2c
13a1 : 91 a9 c8 2c 00 dd 30 fb  66
13a9 : ad 01 dd 91 a9 c8 c0 44  ed
13b1 : d0 e6 f0 ab c9 f1 d0 03  1b
13b9 : 4c 77 c8 c9 f2 d0 03 4c  86
13c1 : 8f c8 a0 00 20 0f c8 aa  cf
13c9 : 30 03 20 c1 c8 c8 c4 6b  78
13d1 : d0 f2 60 a2 05 a5 fb ca  8a
13d9 : dd 4a cc b0 fa bd 4f cc  99
13e1 : 85 6b 60 ff 24 1f 19 12  f8
13e9 : 11 11 12 13 15 00 00 44  44
13f1 : 01 88 02 cc 03 10 05 54  be
13f9 : 06 98 07 dc 08 20 0a 64  1b
1401 : 0b a8 0c ec 0d 30 0f 74  78
1409 : 10 b8 11 fc 12 40 14 84  d6
1411 : 15 c8 16 0c 18 50 19 20  3a
1419 : 3a cc a9 05 20 2c c8 a9  9e
1421 : ff 8d 03 dd a0 00 84 fc  79
1429 : a9 35 85 01 a5 69 a6 6a  03
1431 : 85 a7 86 a8 85 a9 e8 86  97
1439 : aa a0 00 b1 a7 2c 00 dd  01
1441 : 10 fb 8d 01 dd c8 b1 a7  0d
1449 : 2c 00 dd 30 fb 8d 01 dd  df
1451 : c8 d0 e8 b1 a9 2c 00 dd  a9
1459 : 10 fb 8d 01 dd c8 b1 a9  29
1461 : 2c 00 dd 30 fb 8d 01 dd  f7
1469 : c8 c0 44 d0 e6 e6 fc 98  87
1471 : 18 65 a7 85 a7 85 a9 a5  6f
1479 : a8 69 01 85 a8 85 aa e6  f6
1481 : aa a6 fc e4 6b d0 b2 a9  b5
1489 : 37 85 01 60 20 28 c6 93  55
1491 : 1e 20 20 2a 2a 2a 20 4d  1c
1499 : 41 53 54 45 52 20 2d 20  5d
14a1 : 43 4f 50 59 20 50 41 52  f9
14a9 : 41 4c 4c 45 4c 20 56 20  2b
14b1 : 31 2e 33 20 2a 2a 2a 0d  81
14b9 : 0d 20 20 20 28 43 29 20  64
14c1 : 31 39 38 37 20 42 59 20  3e
14c9 : 46 52 41 4e 4b 20 52 49  e4
14d1 : 45 4d 45 4e 53 43 48 4e  e5
14d9 : 45 49 44 45 52 0d 0d 0d  58
14e1 : 20 12 12 41 92 4e 5a 41  3f
14e9 : 48 4c 20 4b 4f 50 49 45  f0
14f1 : 4e 3a 0d 0d 20 12 42 92  02
14f9 : 45 46 45 48 4c 20 53 45  59
1501 : 4e 44 45 4e 0d 0d 20 12  6a
1509 : 44 92 49 52 45 4b 54 4f  d2
1511 : 52 59 20 5a 45 49 47 45  a9
1519 : 4e 0d 0d 20 12 4b 92 4f  9a
1521 : 50 49 45 52 45 4e 0d 0d  c7
1529 : 20 12 56 92 45 52 49 46  d3
1531 : 59 0d 0d 20 12 53 92 54  07
1539 : 41 52 54 54 52 41 43 4b  16
1541 : 3a 0d 0d 20 12 45 92 4e  7b
1549 : 44 54 52 41 43 4b 3a 0d  06
1551 : 0d 20 12 47 92 45 52 41  fb
1559 : 45 54 45 4e 55 4d 4d 45  63
1561 : 52 3a 0d 0d 20 12 50 92  ae
1569 : 52 4f 47 52 41 4d 4d 45  bd
1571 : 4e 44 45 0d 0d 20 12 49  81
1579 : 92 4e 46 4f 0d 00 a2 0f  27
1581 : a0 0d 20 0c e5 a5 fd 20  f5
1589 : 49 c6 a2 11 a0 0b 20 0c  fb
1591 : e5 a5 fe 20 49 c6 a2 13  88
1599 : a0 10 20 0c e5 a5 f9 20  7f
15a1 : 49 c6 a2 05 a0 10 20 0c  ba
15a9 : e5 a5 8f 20 49 c6 a9 00  ba
15b1 : 85 c6 20 8f c2 ea ea c9  56
15b9 : 42 d0 03 4c 86 ce c9 44  3c
15c1 : d0 03 4c c8 ce c9 4b d0  49
15c9 : 06 20 2f c2 4c f3 cc c9  2f
15d1 : 53 d0 0e a9 01 20 3c 03  4d
15d9 : 85 fd 4c 15 ce 85 fd d0  c5
15e1 : 10 c9 45 d0 0f a9 02 20  c8
15e9 : 3c 03 85 fe 4c 15 ce 85  9c
15f1 : fe 4c e5 cd c9 50 d0 03  b1
15f9 : 4c e2 fc c9 47 d0 0e a9  b5
1601 : 03 20 3c 03 85 f9 4c 15  07
1609 : ce 85 f9 d0 e4 c9 41 d0  75
1611 : e0 a9 00 20 3c 03 85 8f  db
1619 : 4c 15 ce 85 8f d0 d2 a9  72
1621 : 0f 20 c3 ff a2 17 a0 01  98
1629 : 20 0c e5 20 28 c6 42 45  19
1631 : 46 45 48 4c 3a 20 00 a9  ad
1639 : 0f a6 f9 a8 20 ba ff a9  5a
1641 : 00 20 bd ff 20 c0 ff a2  0e
1649 : 0f 20 c9 ff 20 cf ff 20  9b
1651 : d2 ff c9 0d d0 f6 20 cc  16
1659 : ff a9 0f 20 c3 ff 4c f3  4a
1661 : cc a2 10 b5 60 9d ef 03  f2
1669 : ca 10 f8 a9 9a 20 d2 ff  a5
1671 : 20 cd bd a9 1f 20 d2 ff  5b
1679 : a9 93 20 d2 ff a9 02 a6  f1
1681 : f9 a0 00 20 ba ff a2 b1  68
1689 : a0 cf a9 01 20 bd ff 20  cc
1691 : c0 ff a2 02 20 c6 ff 20  b2
1699 : cf ff a5 90 d0 73 20 cf  ac
16a1 : ff a9 16 85 8b c6 8b d0  6a
16a9 : 3a 20 28 c6 0d 0d 20 20  d0
16b1 : 20 20 20 57 45 49 54 45  4f
16b9 : 52 20 4d 49 54 20 54 41  b2
16c1 : 53 54 45 4e 44 52 55 43  0c
16c9 : 4b 00 20 cc ff a9 00 85  0e
16d1 : c6 20 e4 ff f0 fb a2 02  5e
16d9 : 20 c6 ff a9 93 20 d2 ff  17
16e1 : 4c 08 cf 20 cf ff 20 cf  46
16e9 : ff f0 26 20 28 c6 0d 20  1b
16f1 : 20 20 00 20 cf ff aa 20  0d
16f9 : cf ff 20 cd bd a9 20 20  73
1701 : d2 ff 20 cf ff f0 9e a4  20
1709 : 90 d0 06 20 d2 ff 4c 69  b8
1711 : cf 20 cc ff a9 02 20 c3  d6
1719 : ff 20 28 c6 0d 0d 20 20  05
1721 : 20 20 20 57 45 49 54 45  bf
1729 : 52 20 4d 49 54 20 54 41  22
1731 : 53 54 45 4e 44 52 55 43  7c
1739 : 4b 00 20 05 c8 a2 10 bd  8a
1741 : ef 03 95 60 ca 10 f8 4c  cd
1749 : f3 cc 24 0a aa bd 54 cc  70
1751 : 18 65 69 85 a7 bd 55 cc  7e
1759 : 65 6a 85 a8 a0 03 b9 f7  63
1761 : cf 91 a7 88 10 f8 a2 40  c8
1769 : a5 a7 18 69 04 85 a7 a5  6b
1771 : a8 69 00 85 a8 a0 04 b9  92
1779 : fb cf 91 a7 88 10 f8 a5  ed
1781 : a7 18 69 05 85 a7 a5 a8  ad
1789 : 69 00 85 a8 ca d0 e6 60  f8
1791 : 55 d4 a5 29 4a 52 94 a5  b4
1799 : 29 50 04 bf fd ff ff a2  08
17a1 : b4 bd ad 17 9d 3b 03 ca  d7
17a9 : d0 f7 4c 55 18 85 a7 aa  d4
17b1 : bd a7 03 85 a9 bd ab 03  f1
17b9 : 85 aa a0 00 b1 a9 09 80  49
17c1 : 91 a9 20 e4 ff f0 fb c9  d6
17c9 : 30 90 f7 a6 a7 dd af 03  42
17d1 : b0 f0 a0 00 91 a9 38 e9  3d
17d9 : 30 85 a8 c8 b1 a9 09 80  9c
17e1 : 91 a9 20 e4 ff f0 fb 48  f3
17e9 : a5 a7 0a 0a 18 65 a7 65  3c
17f1 : a8 aa 68 dd b3 03 90 ea  2f
17f9 : dd c7 03 b0 e5 a0 01 91  1b
1801 : a9 38 e9 30 aa a5 a8 0a  d5
1809 : 0a 0a 85 a7 a5 a8 0a 18  67
1811 : 65 a7 86 a8 18 65 a8 60  11
1819 : d8 65 b3 08 04 06 06 07  28
1821 : 32 35 35 32 31 30 00 00  16
1829 : 00 31 30 30 30 30 31 30  7e
```

```
1831 : 30 30 30 38 30 00 00 00  90
1839 : 3a 31 00 00 00 3a 3a 3a  3b
1841 : 3a 31 3a 3a 3a 3a 31 3a  98
1849 : 32 00 00 00 68 48 20 b2  2a
1851 : cf 68 a8 60 a9 7a a2 18  b4
1859 : 85 fa 86 fb a9 00 a2 fc  9b
1861 : 85 fc 86 fd a2 04 a0 00  92
1869 : b1 fa 91 fc c8 d0 f9 e6  64
1871 : fb e6 fd ca d0 f0 4c 00  7e
1879 : 04 35 06 00 00 7a 03 85  84
1881 : 03 de 03 88 05 00 78 a2  3d
1889 : 45 9a ad 0c 1c 09 0e 8d  66
1891 : 0c 1c ad 00 1c 09 04 8d  4c
1899 : 00 1c 20 5b 06 0a aa bd  f2
18a1 : 00 03 8d 33 03 bd 01 03  15
18a9 : 8d 34 03 4c 00 03 48 38  44
18b1 : e5 22 0a 85 4a 68 85 22  1d
18b9 : a5 4a f0 08 20 2e fa 20  60
18c1 : 29 06 f0 f4 a2 05 a5 22  f5
18c9 : ca dd d4 03 b0 fa bd d9  a5
18d1 : 03 85 43 e0 00 f0 01 ca  a5
18d9 : 8a 0a 0a 0a 0a 0a 85 44  bc
18e1 : ad 00 1c 29 9f 05 44 8d  09
18e9 : 00 1c ad 0c 1c 09 0e 8d  42
18f1 : 0c 1c 60 20 5b 06 85 23  6a
18f9 : 20 35 03 4c 19 03 20 5b  df
1901 : 06 85 23 20 35 03 a9 00  a9
1909 : 85 30 85 34 a9 02 85 31  b1
1911 : a9 00 85 19 a5 19 85 54  21
1919 : a9 08 85 52 a5 23 85 55  a6
1921 : a5 12 45 13 45 23 45 19  38
1929 : 85 53 20 d0 f6 a5 12 85  6a
1931 : 53 a5 13 85 52 a9 00 85  4a
1939 : 54 85 55 20 d0 f6 c6 34  f1
1941 : c6 34 e6 19 a5 19 c5 43  bf
1949 : d0 ca 4c 19 03 ff 24 1f  b4
1951 : 19 12 11 11 12 13 15 a9  3b
1959 : 05 a6 43 86 ad 95 b1 ca  37
1961 : d0 fb 86 b0 a2 2a 20 45  6d
1969 : 06 90 08 a9 f1 20 77 06  f9
1971 : 4c 19 03 50 fe b8 ad 01  83
1979 : 1c c9 52 f0 07 ca d0 e6  04
1981 : a9 f2 d0 e9 a2 03 50 fe  96
1989 : b8 ad 01 1c ca d0 f7 aa  44
1991 : 08 4a 4a 29 1f aa bd c0  36
1999 : f8 28 10 02 09 10 c5 43  98
19a1 : 90 0b a5 b0 e6 b0 c5 43  c8
19a9 : d0 ba 4c 6e 05 aa b5 b2  99
19b1 : f0 f0 30 ee 8a 20 77 06  97
19b9 : a0 00 84 a8 84 a9 84 aa  8c
19c1 : 84 ab 84 b1 84 b0 88 8c  7b
19c9 : 00 18 20 45 06 a0 20 50  0d
19d1 : fe b8 ad 01 1c 8d 01 18  19
19d9 : ee 00 18 45 a8 85 a8 50  70
19e1 : fe b8 ad 01 1c 8d 01 18  29
19e9 : ce 00 18 45 a9 85 a9 50  74
19f1 : fe b8 ad 01 1c 8d 01 18  39
19f9 : ee 00 18 45 aa 85 aa 50  b8
1a01 : fe b8 ad 01 1c 8d 01 18  49
1a09 : ce 00 18 45 ab 85 ab 50  bc
1a11 : fe b8 ad 01 1c 8d 01 18  59
1a19 : ee 00 18 45 b1 85 b1 50  65
1a21 : fe b8 ad 01 1c 8d 01 18  69
1a29 : ce 00 18 45 a8 85 a8 50  a0
1a31 : fe b8 ad 01 1c 8d 01 18  79
1a39 : ee 00 18 45 a9 85 a9 50  e4
1a41 : fe b8 ad 01 1c 8d 01 18  89
1a49 : ce 00 18 45 aa 85 aa 50  e8
1a51 : fe b8 ad 01 1c 8d 01 18  99
1a59 : ee 00 18 45 ab 85 ab 50  2c
1a61 : fe b8 ad 01 1c 8d 01 18  a9
1a69 : ce 00 18 45 b1 85 b1 88  05
1a71 : f0 03 4c 56 04 50 fe b8  f1
1a79 : ad 01 1c 8d 01 18 ee 00  ec
1a81 : 18 45 a8 85 a8 50 fe b8  91
1a89 : ad 01 1c 8d 01 18 ce 00  7c
1a91 : 18 45 a9 85 a9 50 fe b8  f1
1a99 : ad 01 1c 8d 01 18 ee 00  0c
1aa1 : 18 29 40 45 aa 85 aa 50  29
1aa9 : fe ad 01 1c 8d 01 18 ce  21
1ab1 : 00 18 a5 b1 4a 4a ea ea  d5
1ab9 : 45 a8 ea ea 4a 08 ea ea  d1
1ac1 : 4a 45 a9 ea ea 4a 4a 45  2a
1ac9 : aa 29 02 d0 2e 28 6a 4a  05
1ad1 : 45 aa 4a 4a 45 ab 29 10  be
1ad9 : f0 21 b5 b2 09 80 95 b2  6e
1ae1 : c6 ad f0 03 4c eb 03 a9  9e
1ae9 : fd 20 77 06 a2 00 b5 b2  fb
1af1 : 20 77 06 e8 e4 43 d0 f6  05
1af9 : 4c 19 03 d6 b2 f0 e1 d0  49
1b01 : e3 a9 00 85 ac 8d 00 18  d1
1b09 : a5 43 85 ad a9 ce 8d 0c  c6
1b11 : 1c a9 ff 8d 03 1c a9 55  16
1b19 : 8d 01 1c a2 01 20 24 fe  22
1b21 : a9 ff 8d 01 1c a2 05 50  d9
1b29 : fe b8 ca d0 fa a2 08 a4  7e
1b31 : ac b9 00 02 8d 01 1c 50  ec
1b39 : fe b8 c8 ca d0 f3 84 ac  37
1b41 : a2 0b a9 55 8d 01 1c 50  70
1b49 : fe b8 ca d0 fa a9 ff 8d  88
1b51 : 01 1c a2 05 a0 a2 50 fe  08
1b59 : b8 ca d0 fa ad 01 18 ce  eb
1b61 : 00 18 8d 01 1c 50 fe b8  a3
1b69 : ad 01 18 ee 00 18 8d 01  74
1b71 : 1c 50 fe b8 88 d0 e5 a9  86
1b79 : 55 8d 01 1c a2 08 50 fe  02
1b81 : b8 ca d0 fa c6 ad f0 03  d5
1b89 : 4c a7 05 ce 00 18 50 fe  c4
1b91 : b8 50 fe b8 20 00 fe 4c  df
1b99 : 19 03 ea ea ea ea ea ea  d3
1ba1 : ea ea a0 04 84 cb c8 d0  14
1ba9 : fd c6 cb d0 f9 60 a9 00  5f
1bb1 : 8d 03 18 95 00 9d 00 02  6a
1bb9 : e8 d0 f8 4c 22 eb a9 d0  9b
1bc1 : 8d 05 18 38 2c 05 18 10  49
1bc9 : 0a 2c 00 1c 30 f6 ad 01  e0
1bd1 : 1c b8 18 60 a0 00 8c 03  9e
1bd9 : 18 a0 10 2c 00 18 10 fb  c4
1be1 : 8c 00 18 88 2c 00 18 30  08
1be9 : fb ad 01 18 8c 00 18 60  e8
1bf1 : a0 ff 8c 03 18 a0 10 2c  33
1bf9 : 00 18 10 fb 8d 01 18 8c  e3
1c01 : 00 18 88 2c 00 18 30 fb  2e
1c09 : 8c 00 18 60 7f 00 ff 0d  ba
1c11 : 38 79 ff 36 7f b9 33 d9  13
1c19 : ba 10 ff 00 fb bf 39 bf  fd
1c21 : 39 39 ff 00 b9 39 ff f9  50
1c29 : 3d 7f 7f 00 ff 00 ff b9  79
1c31 : 39 b9 ff 00 08 3d 7e ff  ab
1c39 : ff 00 ff 08 ff 00 ff 3c  b2
1c41 : 37 89 ff 00 fb 09 f6 ff  21
1c49 : 39 00 be d2 f9 00 ff 2d  86
1c51 : 3b 39 7f b7 ff 39 dd 76  2e
1c59 : bb 10 ff 00 ff bb 39 f2  c5
1c61 : 39 ea ea a9 15 8d 18 d0  bf
1c69 : a9 01 85 fd 85 8f a9 0f  4d
1c71 : 8d 20 d0 8d 21 d0 a9 08  43
1c79 : 85 52 a5 29 4a 52 94 f5  2b
```

Listing 1. (Schluß)

Diskettenhüllen mit MPS 801

Bringen Sie Ordnung in Ihre Diskettensammlung. »Floppy-Bag 801« schafft dies in zweifacher Hinsicht. Das Programm druckt Diskettenhüllen mit Directoryaufdruck und Diskettenaufkleber.

Wer eine große Diskettensammlung hat, kennt das Problem: Man verliert leicht den Überblick, wo welches Programm gespeichert ist. »Floppy-Bag 801« (Listing), ein auf den MPS 801 zugeschnittenes Programm, hilft Ihnen die Übersicht zu behalten. Laden Sie das Programm mit LOAD "FLOPPY-BAG 801", 8 und starten Sie es mit RUN.

Auf dem Bildschirm erscheint das Hauptmenü. Durch Drücken der Tasten <1> bis <5> können Sie den gewünschten Ausdruck auswählen:

1. Tasche

Es folgt eine Abfrage, ob der Drucker eingeschaltet und das Papier richtig eingespannt ist (das Papier sollte knapp über dem Druckkopf anliegen). Wollen Sie eine Diskettentasche ausdrucken, so geben Sie <J> ein, und der Ausdruck beginnt. Sollen nach erfolgtem Druck weitere Diskettentaschen gedruckt werden, so startet durch die abermalige Eingabe von <J> der Ausdruck erneut.

Durch die Eingabe von <N> wird der Programmlauf beendet.

2. Tasche mit aufgedrucktem Directory und Etikett

Das Programm liest automatisch das Directory der eingelegten Diskette ein. Bei eventuell auftretenden Fehlern seitens des Floppy-Laufwerks wird eine Fehlermeldung ausgegeben. Nach dem Einlesen des Directory werden Sie aufgefordert, folgende Daten einzugeben beziehungsweise mit <RETURN> zu übernehmen:

Datum: Eingabe im Format »tt.mm.jjjj«

Disk-Name: Der aktuelle Diskettenname wird vorgegeben; kann jedoch auch abgeändert werden (max.16 Zeichen).

Disk-NR: Vorgegeben werden 4 Punkte; bis zu 4 verschiedene Zeichen können eingetragen werden (beispielsweise 3/87).

Disk-ID: Vorgegeben wird die aktuelle Disketten-ID; sie kann auch abgeändert werden.

Floppy: Vorgegeben wird »1541«; kann geändert werden.

Commodore: Vorgegeben wird »64«; kann jedoch in einen anderen Typ mit maximal 4 Zeichen geändert werden. Modus: Vorgegeben wird »64«; kann wiederum mit max. 3 Zeichen geändert werden. Sind die Angaben richtig, so können sie mit <J> in den unteren Teil der Bildschirmmaske übernommen werden. Sollten Korrekturen erforderlich sein, so ist <N> zu drücken. Der Cursor springt dann nochmals in die erste Zeile.

Angezeigt wird nun der erste File-Name; durch Drücken der Funktionstasten können die File-Namen bearbeitet werden:

<F1> Filename ändern

Ein neuer File-Name kann eingegeben werden.

<F3> Filename streichen

Der angezeigte File-Name wird nicht ins ausgedruckte Directory übernommen.

<F5> Filename übernehmen

Der angezeigte File-Name wird übernommen.

<F7> Trennung einfügen

Zur besseren Übersichtlichkeit des Directory kann man eine deutliche Trennzeile aus »=«-Zeichen einfügen.

Die Zahl links vor dem aktuellen File-Namen gibt an, wie viele Files bereits übernommen worden sind. Wurden alle File-Namen angezeigt, erscheint die Meldung »Keine Files mehr vorhanden«. Ein beliebiger Tastendruck führt zur schon beschriebenen Druckerabfrage.

Gedruckt wird dabei eine Diskettentasche mit aufgedrucktem Directory. Links oben erscheint fett gedruckt der Disk-Name, rechts oben die Disk-Nummer.

Darunter können in drei Spalten bis zu 39 File-Namen gelistet werden. Zusätzlich zu den File-Namen werden auf dem verbliebenen Platz Linien gezogen. Hier kann man zusätzliche handschriftliche Eintragungen vornehmen.

Unmittelbar unterhalb der Diskettentasche wird der Etikettenaufkleber ausgedruckt. Das Etikett selbst hat eine Größe von zirka 23 x 49 mm und paßt damit auf die gängigsten Selbstklebe-Etiketten.

Das Etikett ist in vier Zeilen aufgeteilt:

1. Zeile: Disk-Nr., Freiraum, Disk-ID
2. Zeile: Diskettenname
3. Zeile: Computer-Modell
4. Zeile: Floppy, Modus

3. Tasche mit aufgedrucktem Directory + Directory für die Rückseite + 2 Etiketten

Der Ablauf entspricht dem Schritt »2«. Sie werden jedoch nach dem ersten Druckvorgang aufgefordert, die Diskette zu wenden.

4. Directory + Etikett

Der Ablauf entspricht Schritt »2« und »3«, Etikett und Directory werden allerdings mit den Maßen 7,1 x 13,3 cm gedruckt. Dieses Directory ist gedacht zum Aufkleben auf eine DIN-A6-Karteikarte oder eine Disk-Tasche (original oder selbstgedruckt).

5. Etikett

Der Ablauf erfolgt wie bisher, jedoch findet keine File-Bearbeitung statt. Gedruckt wird das bekannte Etikett.

Falls eine Änderung der vorgesehenen Daten beabsichtigt ist, kann man jederzeit das Listing modifizieren. In den Zeilen 362 (Vorgaben bei der Datenabfrage) und 476 (Trennzeichen) ist dies ohne weiteres möglich. Ein sinnvolles Ausnutzen des Freiraumes auf der Etikette läßt sich in Zeile 700 bewerkstelligen. REM-Zeilen können jederzeit gelöscht werden, sie werden im Programmablauf nicht angesprungen. Um die Arbeit mit Floppy-Bag 801 zu beschleunigen, empfiehlt es sich, das Programm zu compilieren.

(J. Gubesch/sk)

```
100 REM********************************     <150>
102 REM*  DISKHUELLEN MPS 801          *     <152>
104 REM********************************     <154>
106 REM* EIN PROGRAMM ZUR HERSTELLUNG *     <251>
108 REM*  NEUER DISKETTENTASCHEN UND  *     <052>
110 REM*     AUFKLEBE-ETIKETTEN       *     <156>
112 REM********************************     <162>
124 REM*       JOHANN GUBESCH         *     <216>
126 REM*       HENGTEWEG 12           *     <015>
128 REM*       4420 COESFELD          *     <119>
130 REM*       02541/81472            *     <131>
132 REM********************************     <182>
134 GOSUB 160                               <174>
136 GOSUB 190                               <224>
138 IF Y=1 THEN GOSUB 500:GOSUB 530:GOSUB
    796                                     <103>
140 DIM DB$(144):AX$="{20SPACE}" :DIM NA$(
    144):XY=0                               <142>
142 GOSUB 254                               <046>
144 IF Y=2 THEN GOSUB 344:GOSUB 428:GOSUB
    500:GOSUB 578:GOSUB 796                 <253>
146 IF Y=3 THEN GOSUB 344:GOSUB 428:GOSUB
    500:GOSUB 578:XY=1                      <133>
148 IF Y=3 THEN GOSUB 254:GOSUB 344         <110>
150 IF Y=3 THEN GOSUB 428:GOSUB 500:GOSUB
    726:GOSUB 796                           <200>
152 IF Y=4 THEN GOSUB 344:GOSUB 428:GOSUB
    500:GOSUB 726:GOSUB 796                 <061>
154 IF Y=5 THEN GOSUB 344:GOSUB 500:GOSUB
    672:GOSUB 796                           <227>
156 END                                     <158>
158 REM ---- MODUL RAHMEN -------------     <252>
160 POKE 53281,1:POKE 53280,1:PRINT CHR$(1
    42)                                     <229>
162 PRINT"{CLR,RVSON,RED,9SPACE}**{2SPACE}
    FLOPPY-BAG 801{2SPACE}**{9SPACE}";      <014>
164 PRINT"{RED,7SPACE}FUER MPS 801 UND KOM
    PATIBLE{6SPACE}";                       <103>
166 PRINT"{RED,SPACE}VON J.GUBESCH, COESFE
    LD, FEBRUAR 1987{2SPACE,RVOFF}"         <136>
168 POKE 646,0:PRINT"{UP}";                 <237>
170 FOR A=1 TO 16                           <002>
172 PRINT"_{38SPACE}_";                     <224>
174 NEXT                                    <184>
176 PRINT"Q***********R*****************R**
    R*******W";                             <224>
178 PRINT"_{11SPACE}_{16SPACE}_{2SPACE}_{6
    SPACE}_";                               <051>
180 PRINT"Q***********ER*****************E*R
    E*******W";                             <068>
182 PRINT"_{12SPACE}_{17SPACE}_{7SPACE}_"; <194>
184 PRINT"Z************E******************E
    *******X{22UP}"                         <240>
186 RETURN                                  <244>
188 REM ---- MODUL AUSWAHLMENUE -------     <124>
190 PRINT:PRINT                             <142>
192 PRINT "{4UP,4RIGHT,GREEN,RVSON,32SPACE
    ,RVOFF}"                                <057>
194 PRINT     "{UP,4RIGHT,RVSON,3SPACE}WAE
    HLEN SIE BITTE DEN GE -{3SPACE}"        <031>
196 PRINT      "{4RIGHT,RVSON,6SPACE}WUENS
    CHTEN AUSDRUCK{7SPACE}"                 <056>
198 PRINT     "{UP,4RIGHT,RVSON,12SPACE,RV
    OFF}( 1-5 ){RVSON,13SPACE}"             <137>
200 PRINT      "{4RIGHT,RVSON,32SPACE}"     <222>
202 PRINT"{BLUE,2UP}"                       <217>
204 PRINT"{4RIGHT,RVSON}1{RVOFF}- TASCHE"  <239>
206 PRINT"{UP,4RIGHT,RVSON}2{RVOFF}- TASCH
    E MIT AUFGEDRUCKTEM DI-"                <119>
208 PRINT"{4RIGHT,3SPACE}RECTORY + ETIKETT
    "                                       <180>
210 PRINT"{UP,4RIGHT,RVSON}3{RVOFF}- TASCH
    E MIT AUFGEDRUCKTEM DI-"                <127>
212 PRINT"{4RIGHT,3SPACE}RECTORY + DIRECTO
    RY FUER DIE "                           <114>
214 PRINT"{4RIGHT,UP,3SPACE}RUECKSEITE + 2
     ETIKETTEN"                             <173>
216 PRINT"{4RIGHT,RVSON}4{RVOFF}- DIRECTOR
    Y + ETIKETT"                            <182>
218 PRINT"{UP,4RIGHT,RVSON}5{RVOFF}- ETIKE
    TT"                                     <168>
220 GET AN$:IF AN$=""THEN 220               <035>
222 IF AN$="1" THEN Y=1:GOSUB 238:GOTO 234 <202>
224 IF AN$="2" THEN Y=2:GOSUB 238:GOTO 234 <237>
226 IF AN$="3" THEN Y=3:GOSUB 238:GOTO 234 <016>
228 IF AN$="4" THEN Y=4:GOSUB 238:GOTO 234 <051>
230 IF AN$="5" THEN Y=5:GOSUB 238:GOTO 234 <086>
232 GOTO 220                                <200>
234 RETURN                                  <036>
236 REM ---- MODUL BEDIENERFELD FREI --     <147>
238 PRINT"{HOME}"                           <218>
240 PRINT"{DOWN}"                           <188>
242 FOR T= 1 TO 7                           <175>
244 PRINT"{RIGHT,38SPACE}"                  <128>
246 PRINT"{UP,RIGHT,38SPACE}"               <177>
248 NEXT                                    <002>
250 PRINT"{RIGHT,38SPACE}"                  <134>
252 RETURN                                  <054>
254 REM ---- MODUL DIRECTORY LESEN ----     <133>
256 PRINT"{HOME,GREEN,2DOWN}"      :AW$="{R
    VSON,32SPACE}"                          <217>
258 PRINT"{4RIGHT}";AW$                     <093>
```

```
260 PRINT"{UP,4RIGHT}";AW$                              <094>
262 PRINT"{4RIGHT}";AW$                                 <097>
264 PRINT"{UP,4RIGHT}";AW$                              <098>
266 PRINT"{4RIGHT}";AW$                                 <101>
268 PRINT"{5UP,8RIGHT,RVSON,2SPACE}EINEN M
    OMENT BITTE -"                                      <237>
270 PRINT"{4RIGHT,RVSON}"                               <038>
272 PRINT"{UP,4RIGHT,RVSON,2SPACE}ICH LESE
     DAS DIRECTORY EIN !"                               <133>
274 IF XY=1 THEN GOSUB 320:GOTO 256                     <122>
276 I=0:GB=1:CLOSE 15:OPEN 15,8,15:CLOSE 1
    :OPEN 1,8,0,"$0"                                    <155>
278 INPUT#15,T,K$,O,O:IF T<>0 THEN PRINT"{
    RIGHT,3UP,4RIGHT,RVSON}FLOPPYFEHLER:{R
    VOFF}";                                             <125>
280 IF T<>0 THEN PRINT"{RVSON,2SPACE}";K$:
    PRINT"{RVSON,9RIGHT,RVOFF}FLOPPY EIN -
     DISK REIN{RVSON,3SPACE}"                           <152>
282 IF T<>0 THEN PRINT "{UP,5RIGHT,RVSON,8
    SPACE,RVOFF}TASTE DRUECKEN{RVSON,9SPAC
    E}"                                                 <153>
284 IF T<>0 THEN POKE 198,0:WAIT 198,1:RUN
     134                                                <142>
286 GET#1,DA$,DE$                                       <001>
288 GET#1,DA$,DE$:DB$(I+1)="":GET#1,DA$,DE
    $                                                   <082>
290 C=0                                                 <201>
292 IF DA$<>"" THEN C=ASC(DA$)                          <076>
294 IF DE$<>"" THEN C=C+ASC(DE$)*256                    <059>
296 FR$=MID$(STR$(C),2)                                 <239>
298 GET#1,DE$:IF DE$="" THEN ZA=I:GOTO 316              <138>
300 IF DE$<>CHR$(34) THEN 298                           <132>
302 GET#1,DE$:IF DE$<>CHR$(34) THEN DB$(I)
    =DB$(I)+DE$:GOTO 302                                <022>
304 GET#1,DE$:IF DE$=CHR$(32) THEN 304                  <016>
306 DC$=""                                              <144>
308 DC$=DC$+DE$:GET#1,DE$:IF DE$<>"" THEN
    308                                                 <162>
310 IF GB=1 THEN ID$=LEFT$(DC$,2):GB=0                  <163>
312 DB$(I)=LEFT$(DB$(I)+AX$,17)                         <001>
314 I=I+1:GOTO 288                                      <214>
316 CLOSE 1:CLOSE 15                                    <230>
318 RETURN                                              <122>
320 PRINT"{HOME,GREEN,2DOWN}"       :AW$="{R
    VSON,32SPACE}"                                      <025>
322 PRINT"{4RIGHT}";AW$                                 <157>
324 PRINT"{UP,4RIGHT}";AW$                              <158>
326 PRINT"{4RIGHT}";AW$                                 <161>
328 PRINT"{UP,4RIGHT}";AW$                              <162>
330 PRINT"{4RIGHT}";AW$                                 <165>
332 PRINT"{5UP,8RIGHT,RVSON}BITTE DISKETTE
     WENDEN "                                           <215>
334 PRINT"{4RIGHT,RVSON}"                               <102>
336 PRINT"{UP,4RIGHT,RVSON,11SPACE,RVOFF,S
    PACE}TASTE{SPACE,RVSON,8SPACE}"                     <018>
338 POKE 198,0:WAIT 198,1:XY=0                          <170>
340 RETURN                                              <144>
342 REM ---- MODUL DATENEINGABE -------                 <187>
344 PRINT"{HOME,GREEN,2DOWN}"       :AW$="{R
    VSON,32SPACE}"                                      <049>
346 PRINT"{4RIGHT}";AW$                                 <181>
348 PRINT"{UP,4RIGHT}";AW$                              <182>
350 PRINT"{4RIGHT}";AW$                                 <185>
352 PRINT"{UP,4RIGHT}";AW$                              <186>
354 PRINT"{4RIGHT}";AW$                                 <189>
356 PRINT"{5UP,6RIGHT,RVSON,2SPACE}BITTE G
    EBEN SIE FOLGENDE"                                  <005>
358 PRINT"{4RIGHT,RVSON,SPACE}DATEN EIN BZ
    W. UEBERNEHMEN MIT"                                 <160>
360 PRINT"{UP,6RIGHT,RVSON,10SPACE,RVOFF}R
    ETURN"                                              <254>
362 CO$=" 64 ":MO$=" 64":FL$="1541":NR$=".
    ..."                                                <088>
364 POKE 19,1:GOSUB 1000:POKE 19,0                      <030>
366 TE$=LEFT$(TE$,10)                                   <251>
368 HE$=LEFT$(DB$(0),16)                                <135>
370 PRINT"{UP,6RIGHT,RVSON}DISK-NAME{2SPAC
    E}:{RVOFF,SPACE}";HE$;:POKE 19,1:INPUT
    "{16LEFT}";HE$                                      <252>
372 POKE 19,0                                           <129>
374 HE$=LEFT$(HE$,16)                                   <135>
376 PRINT "{DOWN,5RIGHT,RVSON}DISK-NUMMER:
    {RVOFF,SPACE}";NR$;:POKE 19,1:INPUT"{4
    LEFT}";NR$                                          <024>
378 POKE 19,0                                           <135>
380 NR$=LEFT$(NR$,4 )                                   <114>
382 PRINT "{DOWN,2RIGHT,UP,3RIGHT,RVSON}DI
    SK-ID{4SPACE}:{RVOFF,SPACE}";ID$;:POKE
     19,1:INPUT"{2LEFT}";ID$                            <165>
384 POKE 19,0                                           <141>
386 ID$=LEFT$(ID$,2 )                                   <179>
388 PRINT "{DOWN,5RIGHT,RVSON}FLOPPY{5SPAC
    E}:{RVOFF,SPACE}";FL$;:POKE 19,1:INPUT
    "{4LEFT}";FL$                                       <105>
390 POKE 19,0                                           <147>
392 FL$=LEFT$(FL$,4)                                    <026>
394 PRINT "{DOWN,UP,5RIGHT,RVSON}COMMODORE
    {2SPACE}:{RVOFF,SPACE}";CO$;:POKE 19,1
    :INPUT"{4LEFT}";CO$                                 <110>
396 POKE 19,0                                           <153>
398 CO$=LEFT$(CO$,4)                                    <047>
400 PRINT "{DOWN,5RIGHT,RVSON}MODUS{6SPACE
    }:{RVOFF,SPACE}";MO$;:POKE 19,1:INPUT"
    {3LEFT}";MO$                                        <204>
402 POKE 19,0                                           <159>
404 MO$=LEFT$(MO$,3)                                    <087>
406 PRINT "{DOWN,5RIGHT}STIMMEN ALLE ANGAB
    EN ? (J/N)"                                         <186>
408 GET AN$:IF AN$="" THEN 408                          <240>
410 IF AN$="J" THEN 418                                 <224>
412 IF AN$="N" THEN PRINT:PRINT"{4UP,6RIGH
    T,30SPACE}"                                         <169>
414 IF AN$="N" THEN  PRINT"{12UP}";:GOTO 3
    64                                                  <173>
416 GOTO 408                                            <115>
418 PRINT"{RIGHT}";TE$;"{2RIGHT}";HE$;"{RI
    GHT}";ID$;"{2RIGHT}";NR$;"{DOWN,3RIGHT
    }FLOPPY{2SPACE}";FL$;"{2RIGHT}";                    <128>
420 PRINT" COMMODORE ";CO$;"{3RIGHT}";MO$;
    "'"                                                 <068>
422 GOSUB 238                                           <169>
424 RETURN                                              <228>
426 REM ---- MODUL FILES BEARBEITEN ---                 <218>
428 PRINT"{HOME,GREEN,2DOWN}"       :AW$="{R
    VSON,32SPACE}"                                      <133>
430 PRINT"{4RIGHT}";AW$                                 <009>
432 PRINT"{UP,4RIGHT}";AW$                              <010>
434 PRINT"{4RIGHT}";AW$                                 <013>
436 PRINT"{UP,4RIGHT}";AW$                              <014>
438 PRINT"{4RIGHT}";AW$                                 <017>
440 PRINT  "{7UP,2DOWN,7RIGHT}F1{RVSON}-
    FILENAME AENDERN"                                   <166>
442 PRINT"{7RIGHT}F3{RVSON}- FILENAME STRE
    ICHEN"                                              <169>
444 PRINT"{UP,7RIGHT}F5{RVSON}- FILENAME U
    EBERNEHMEN"                                         <103>
446 PRINT"{7RIGHT}F7{RVSON}- TRENNUNG EINF
    UEGEN"                                              <061>
448 PRINT"{UP,4RIGHT,RVSON,32SPACE}"                    <147>
450 X1=1:X=0:T=X                                        <139>
452 IF ZA>39 THEN ZA = 39                               <199>
454 FOR T=0 TO ZA-2                                     <087>
456 PRINT"{BLUE,3DOWN,8RIGHT}";X;"{2SPACE}
    ";LEFT$(DB$(X+X1),17);"{19LEFT,4UP}"                <134>
458 GET AN$:IF AN$="" THEN 458                          <039>
460 IF AN$="{F5}" THEN NA$(X)=LEFT$(DB$(X+
    X1),17):GOTO 484                                    <113>
462 IF AN$="{F3}" THEN X=X-1:X1=X1+1                    <063>
464 IF AN$="{F3}" AND X<0 THEN X=-1                     <156>
466 IF AN$="{F3}" THEN 484                              <067>
468 IF AN$="{F1}" THEN PRINT"{6DOWN,4RIGHT
    ,RVSON}NEUER NAME:{RVOFF,SPACE}.";:POK
    E 19,1                                              <006>
470 IF AN$="{F1}" THEN INPUT"{LEFT}";NA$(X
    ):POKE 19,0:NA$(X)=LEFT$(NA$(X)     ,17
    )                                                   <068>
472 IF AN$="{F1}" THEN PRINT "{2UP,2RIGHT,
    35SPACE}"                                           <017>
474 IF AN$="{F1}" THEN PRINT"{6UP}";:GOTO
    484                                                 <013>
476 IF AN$="{F7}" THEN NA$(X)="==========
    ===== "                                             <002>
478 IF AN$="{F7}" THEN X1=X1-1:T=T-1                    <078>
480 IF AN$="{F7}" THEN GOTO 484                         <076>
482 GOTO 458                                            <005>
484 X=X+1:IF X=38 THEN T=ZA-2                           <189>
486 NEXT                                                <242>
488 PRINT  "{3DOWN,5RIGHT,SPACE}KEINE FILE
    S MEHR VORHANDEN"                                   <117>
490 FOR T=X TO 38:NA$(T)="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
    ":NEXT                                              <220>
492 POKE 198,0:WAIT 198,1                               <210>
494 GOSUB 238                                           <241>
```

Listing 1. (Fortsetzung)

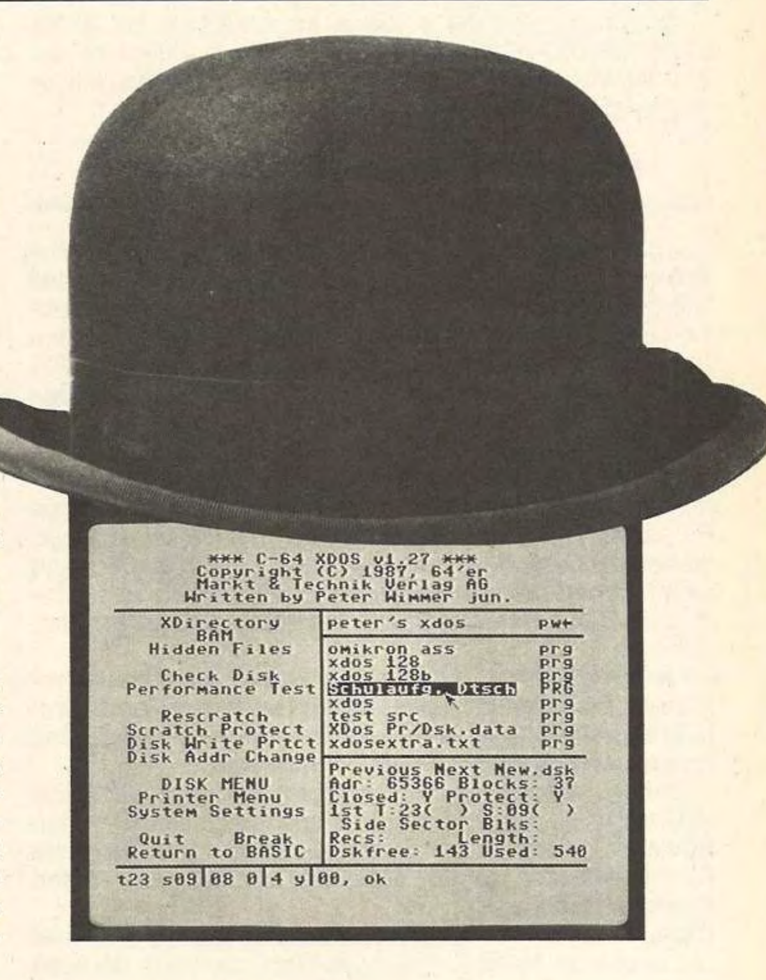
```
496 RETURN                                    <044>
498 REM ---- MODUL DRUCKVORBEREITUNG --        <249>
500 PRINT"{HOME,GREEN,2DOWN}"       :AW$="{R
    VSON,32SPACE}"                            <205>
502 PRINT"{4RIGHT}";AW$                       <081>
504 PRINT"{UP,4RIGHT}";AW$                    <082>
506 PRINT"{4RIGHT}";AW$                       <085>
508 PRINT"{UP,4RIGHT}";AW$                    <086>
510 PRINT"{4RIGHT}";AW$                       <089>
512 PRINT"{7UP,2DOWN,7RIGHT,RVSON}DRUCKER
    EINGESCHALTET UND"                        <211>
514 PRINT"{6RIGHT,RVSON}PAPIER RICHTIG EIN
    GESPANNT ?"                               <067>
516 PRINT"{UP,16RIGHT}(J/N)"                  <230>
518 GET AN$:IF AN$="" THEN 518                <225>
520 IF AN$="J" THEN 524                       <231>
522 GOTO 518                                  <247>
524 GOSUB 238                                 <017>
526 RETURN                                    <076>
528 REM ---- MODUL DR HUELLE ----------       <166>
530 PRINT"{HOME,GREEN,2DOWN}"       :AW$="{R
    VSON,32SPACE}"                            <237>
532 PRINT"{4RIGHT}";AW$                       <113>
534 PRINT"{UP,4RIGHT}";AW$                    <114>
536 PRINT"{4RIGHT}";AW$                       <117>
538 PRINT"{UP,4RIGHT}";AW$                    <118>
540 PRINT"{4RIGHT}";AW$                       <121>
542 PRINT"{5UP,12RIGHT,RVSON}AUF GEHT'S -"    <032>
544 PRINT"{4RIGHT,RVSON}"                     <139>
546 PRINT"{UP,10RIGHT,RVSON}JETZT WIRD GED
    RUCKT !"                                  <177>
548 PRINT"{5DOWN,11RIGHT,BLUE}DISKETTEN-TA
    SCHE"                                     <221>
550 OPEN 4,4                                  <145>
552 PRINT#4,"U-------------------------------
    --------------------------------I"        <144>
554 FOR T=1 TO 22                             <155>
556 PRINT#4,"G{4SPACE}:{54SPACE}:{4SPACE}M
    "                                         <208>
558 NEXT                                      <060>
560 PRINT#4,"J----I.........................
    ................................U----K"   <082>
562 FOR T=1 TO 28                             <169>
564 PRINT#4,"{5SPACE}_{54SPACE}_"             <200>
566 NEXT                                      <068>
568 PRINT#4,"{5SPACE}J-----------------------
    -------------------------------K"         <236>
570 CLOSE 4                                   <097>
572 GOSUB 238                                 <065>
574 RETURN                                    <124>
576 REM ---- MODUL DR HUELLE+DIR+ETI --       <223>
578 PRINT"{HOME,GREEN,2DOWN}":AW$="{RVSON,
    32SPACE}"                                 <029>
580 PRINT"{4RIGHT}";AW$                       <161>
582 PRINT"{UP,4RIGHT}";AW$                    <162>
584 PRINT"{4RIGHT}";AW$                       <165>
586 PRINT"{UP,4RIGHT}";AW$                    <166>
588 PRINT"{4RIGHT}";AW$                       <169>
590 PRINT"{5UP,13RIGHT,RVSON}AUF GEHT'S -"    <020>
592 PRINT"{4RIGHT,RVSON}"                     <106>
594 PRINT"{UP,10RIGHT,RVSON}JETZT WIRD GED
    RUCKT !"                                  <225>
596 PRINT"{BLUE,5DOWN,3RIGHT}DISKETTEN-TAS
    CHE MIT AUFGEDRUCKTEM"                    <094>
598 PRINT"{5RIGHT}DIRECTORY UND AUFKLEBEET
    IKETT"                                    <070>
600 CLOSE 1:OPEN 1,4                          <147>
602 PRINT#1,CHR$(145)                         <231>
604 PRINT#1,"U-------------------------------
    --------------------------------I         <138>
606 PRINT#1,"G{4SPACE}:{54SPACE}:{4SPACE}M
    "                                         <234>
608 PRINT#1,"G{4SPACE}:  "CHR$(14);HE$;CHR$
    (15);                                     <138>
610 PRINT#1,"{4SPACE}DISK-NR:";CHR$(14);NR
    $;CHR$(15);" :{4SPACE}M"                  <188>
612 PRINT#1,"G{4SPACE}:  ";:FOR I=1 TO 52:P
    RINT#1,"C";:NEXT:PRINT#1," :{4SPACE}M"    <042>
614 PRINT#1,"G{4SPACE}:  "NA$(0);" ";NA$(13
    );" ";NA$(26); ":{4SPACE}M"               <230>
616 PRINT#1,"G{4SPACE}:  "NA$(1);" ";NA$(14
    );" ";NA$(27); ":{4SPACE}M"               <017>
618 PRINT#1,"G{4SPACE}:  "NA$(2);" ";NA$(15
    );" ";NA$(28); ":{4SPACE}M"               <060>
620 PRINT#1,"G{4SPACE}:  "NA$(3);" ";NA$(16
    );" ";NA$(29); ":{4SPACE}M"               <103>
622 PRINT#1,"G{4SPACE}:  "NA$(4);" ";NA$(17
    );" ";NA$(30); ":{4SPACE}M"               <070>
624 PRINT#1,"G{4SPACE}:  "NA$(5);" ";NA$(18
    );" ";NA$(31); ":{4SPACE}M"               <113>
626 PRINT#1,"G{4SPACE}:  "NA$(6);" ";NA$(19
    );" ";NA$(32); ":{4SPACE}M"               <156>
628 PRINT#1,"G{4SPACE}:  "NA$(7);" ";NA$(20
    );" ";NA$(33); ":{4SPACE}M"               <062>
630 PRINT#1,"G{4SPACE}:  "NA$(8);" ";NA$(21
    );" ";NA$(34); ":{4SPACE}M"               <106>
632 PRINT#1,"G{4SPACE}:  "NA$(9);" ";NA$(22
    );" ";NA$(35); ":{4SPACE}M"               <149>
634 PRINT#1,"G{4SPACE}:  "NA$(10);" ";NA$(2
    3);" ";NA$(36); ":{4SPACE}M"              <212>
636 PRINT#1,"G{4SPACE}:  "NA$(11);" ";NA$(2
    4);" ";NA$(37); ":{4SPACE}M"              <040>
638 PRINT#1,"G{4SPACE}:  "NA$(12);" ";NA$(2
    5);" ";NA$(38); ":{4SPACE}M"              <124>
640 PRINT#1,"G{4SPACE}:  ";:FOR I=1 TO 52:P
    RINT#1,"C";:NEXT:PRINT#1," :{4SPACE}M"    <070>
642 FR$=LEFT$(FR$+"{3SPACE}",3)               <017>
644 PRINT#1,"G{4SPACE}: STAND:"TE$"{20SPAC
    E}"FR$"{2SPACE}BLOCKS FREE :{4SPACE}M"    <187>
646 PRINT#1,"G{4SPACE}:{54SPACE}:{4SPACE}M    <138>
648 PRINT#1,"G{4SPACE}:{54SPACE}:{4SPACE}M    <140>
650 PRINT#1,"G{4SPACE}:{54SPACE}:{4SPACE}M    <142>
652 PRINT#1,"G{4SPACE}:{54SPACE}:{4SPACE}M    <144>
654 PRINT#1,"J----I.........................
    ................................U----K    <118>
656 FOR T=1 TO 28                             <007>
658 PRINT#1,"{5SPACE}_{54SPACE}_"             <014>
660 NEXT                                      <162>
662 PRINT#1,"{5SPACE}J-----------------------
    -------------------------------K"         <050>
664 PRINT CHR$(10)CHR$(10)CHR$(10)            <146>
666 GOSUB 696:                                <108>
668 RETURN                                    <218>
670 REM ---- MODUL DR ETIKETT ---------       <196>
672 PRINT"{HOME,GREEN,2DOWN}":AW$="{RVSON,
    32SPACE}"                                 <123>
674 PRINT"{4RIGHT}";AW$                       <255>
676 PRINT"{UP,4RIGHT}";AW$                    <000>
678 PRINT"{4RIGHT}";AW$                       <003>
680 PRINT"{UP,4RIGHT}";AW$                    <004>
682 PRINT"{4RIGHT}";AW$                       <007>
684 PRINT"{5UP,13RIGHT,RVSON}AUF GEHT'S -"    <114>
686 PRINT"{4RIGHT,RVSON}"                     <200>
688 PRINT"{UP,10RIGHT,RVSON}JETZT WIRD GED
    RUCKT !"                                  <063>
690 PRINT"{5DOWN,11RIGHT,BLUE}DISKETTEN-ET
    IKETT"                                    <169>
692 CLOSE 1:OPEN 1,4                          <239>
694 PRINT#1,CHR$(145)                         <067>
696 PRINT#1,CHR$(15)"U*********R*******R**I"
    CHR$(8)                                   <081>
698 IF NR$=""THEN NR$="{4SPACE}"              <049>
700 PRINT#1,CHR$(15)"_"CHR$(14)NR$CHR$(15)
    "_{6SPACE}_"ID$"_"CHR$(8)                 <146>
702 PRINT#1,CHR$(15)"Q*********E*******E**W"
    CHR$(8)                                   <104>
704 PRINT#1,CHR$(15)"_ "HE$" _"CHR$(8)        <012>
706 PRINT#1,CHR$(15)"Q*********************W"
    CHR$(8)                                   <214>
708 CO$="COMMODORE{2SPACE}"+CO$               <139>
710 PRINT#1,CHR$(15)"_{2SPACE}"CHR$(17)CO$
    CHR$(15)" _"CHR$(8)                       <028>
712 PRINT#1,CHR$(15)"Q****R****************W"
    CHR$(8)                                   <206>
714 MO$=MO$+"'ER MODUS"                       <077>
716 PRINT#1,CHR$(15)"_"FL$"_"CHR$(17)MO$"
    _"CHR$(8)                                 <011>
718 PRINT#1,CHR$(15)"J****E****************K"
    CHR$(15)                                  <168>
720 GOSUB 238                                 <213>
722 RETURN                                    <016>
724 REM ---- MODUL DR DIR+ET ----------       <138>
726 PRINT"{HOME,GREEN,2DOWN}":AW$="{RVSON,
    32SPACE}"                                 <177>
728 PRINT"{4RIGHT}";AW$                       <053>
730 PRINT"{UP,4RIGHT}";AW$                    <054>
732 PRINT"{4RIGHT}";AW$                       <057>
734 PRINT"{UP,4RIGHT}";AW$                    <058>
736 PRINT"{4RIGHT}";AW$                       <061>
738 PRINT"{5UP,13RIGHT,RVSON}AUF GEHT'S -"    <168>
740 PRINT"{4RIGHT,RVSON}"                     <254>
742 PRINT"{UP,10RIGHT,RVSON}JETZT WIRD GED
    RUCKT !"                                  <117>
744 PRINT"{2DOWN,RIGHT,2DOWN,5RIGHT,BLUE}D
```

```
    IRECTORY + AUFKLEBEETIKETT"                         <253>
746 CLOSE 1:OPEN 1,4                                    <037>
748 PRINT#1,CHR$(145)                                   <121>
750 PRINT#1,"{3SPACE}"CHR$(14);HE$;CHR$(15
    );"{4SPACE}DISK-NR:";CHR$(14);NR$;CHR$
    (15)                                                <134>
752 PRINT#1,"{3SPACE}";:FOR I=1 TO 52:PRIN
    T#1,"£";:NEXT:PRINT#1                               <091>
754 PRINT#1,"{3SPACE}"NA$(0);"  ";NA$(13);"
    ";NA$(26)                                           <034>
756 PRINT#1,"{3SPACE}"NA$(1);"  ";NA$(14);"
    ";NA$(27)                                           <111>
758 PRINT#1,"{3SPACE}"NA$(2);"  ";NA$(15);"
    ";NA$(28)                                           <187>
760 PRINT#1,"{3SPACE}"NA$(3);"  ";NA$(16);"
    ";NA$(29)                                           <007>
762 PRINT#1,"{3SPACE}"NA$(4);"  ";NA$(17);"
    ";NA$(30)                                           <064>
764 PRINT#1,"{3SPACE}"NA$(5);"  ";NA$(18);"
    ";NA$(31)                                           <141>
766 PRINT#1,"{3SPACE}"NA$(6);"  ";NA$(19);"
    ";NA$(32)                                           <217>
768 PRINT#1,"{3SPACE}"NA$(7);"  ";NA$(20);"
    ";NA$(33)                                           <197>
770 PRINT#1,"{3SPACE}"NA$(8);"  ";NA$(21);"
    ";NA$(34)                                           <017>
772 PRINT#1,"{3SPACE}"NA$(9);"  ";NA$(22);"
    ";NA$(35)                                           <093>
774 PRINT#1,"{3SPACE}"NA$(10);"  ";NA$(23);
    "  ";NA$(36)                                        <113>
776 PRINT#1,"{3SPACE}"NA$(11);"  ";NA$(24);
    "  ";NA$(37)                                        <008>
778 PRINT#1,"{3SPACE}"NA$(12);"  ";NA$(25);
    "  ";NA$(38)                                        <158>
780 PRINT#1,"{3SPACE}";:FOR I=1 TO 52:PRIN
    T#1,"£";:NEXT:PRINT#1                               <121>
782 PRINT#1,"{3SPACE}STAND:"TE$"{20SPACE}"
    ;FR$;"{2SPACE}BLOCKS FREE"                          <061>
784 PRINT#1:PRINT#1:CLOSE 1                             <034>
786 CLOSE 1:OPEN 1,4                                    <079>
788 PRINT#1,CHR$(145)                                   <163>
790 GOSUB 696                                           <092>
792 RETURN                                              <088>
794 REM ---- MODUL NOCHMAL ------------                 <165>
796 PRINT"{HOME,GREEN,2DOWN}":AW$="{RVSON,
    32SPACE}"                                           <249>
798 PRINT"{4RIGHT}";AW$                                 <125>
800 PRINT"{UP,4RIGHT}";AW$                              <044>
802 PRINT"{4RIGHT}";AW$                                 <129>
804 PRINT"{UP,4RIGHT}";AW$                              <130>
806 PRINT"{4RIGHT}";AW$                                 <133>
808 PRINT"{5UP,5RIGHT,RVSON}WOLLEN SIE NOC
    H MEHR DRUCKEN ?"                                   <099>
810 PRINT"{4RIGHT,RVSON}"                               <070>
812 PRINT"{UP,16RIGHT}(J/N)"                            <016>
814 GET AN$:IF AN$="" THEN 814                          <133>
816 IF AN$="J" THEN RUN 134                             <153>
818 IF AN$="N" THEN PRINT"{BLUE,2DOWN,5RIG
    HT}HOFFENTLICH SIND DIE AUSDRUCKE"                  <000>
820 IF AN$="N" THEN PRINT"{5RIGHT}ZU IHRER
    ZUFRIEDENHEIT AUS-"                                 <049>
822 IF AN$="N" THEN PRINT"{5RIGHT}AUSGEFAL
    LEN{2SPACE}!!"                                      <091>
824 IF AN$="N" THEN PRINT"{21RIGHT,UP,RED}
    TSCHUESS - "                                        <037>
826 IF AN$="N" THEN PRINT"{21RIGHT}BIS BAL
    D !!"                                               <227>
828 IF AN$="N" THEN POKE 198,0:WAIT 198,1:
    PRINT"{CLR}":RETURN                                 <058>
1000 PRINT"{BLUE,2DOWN,6RIGHT,RVSON}DATUM{
     6SPACE}:{RVOFF,SPACE}TT.MM.JJJJ{10LEF
     T}":ZE=10:SP=19                                    <118>
1005 FOR DA=1 TO 8                                      <091>
1010 GET DA$:IF DA$=""THEN 1010                         <175>
1020 TE$=TE$+DA$                                        <162>
1040 POKE 781,ZE:POKE 782,SP:SYS 58634                  <115>
1050 PRINT DA$:SP=SP+1                                  <205>
1060 IF SP=21 THEN SP=22:TE$=TE$+"."                    <181>
1070 IF SP=24 THEN SP=25:TE$=TE$+"."                    <161>
1090 NEXT                                               <084>
1100 RETURN                                             <142>
```

Listing 1. »Floppy-Bag 801«. Bitte mit dem Checksummer auf Seite 158 eingeben.

XDOS – alles unter einem Hut

Alle Funktionen zur Floppy-Bedienung, die sowohl für den Programmierer als auch für den Anwender interessant sind, vereinigt XDOS in sich. XDOS ist mehr als ein reines Disketten-Utility, sondern stellt mit seiner komfortablen Benutzeroberfläche schon fast ein eigenes Betriebssystem dar.



Von der äußeren Form her und durch die Bedienung mit Maus oder Joystick, ähnelt XDOS durch seine Window-Technik den Benutzeroberflächen eines Atari ST oder auch eines Amiga. Einen Überblick über die Funktionen von XDOS finden Sie in Tabelle 1. Für die Bedienung ist unbedingt ein Joystick oder eine Maus erforderlich, ein Reset-Schalter ist ebenso von großem Nutzen, da das Programm am einfachsten mit Hilfe eines Reset gestartet wird. Unbedingt benötigt wird auch ein Drucker, da viele Menüpunkte keine Bildschirm-, sondern nur eine Druckerausgabe besitzen.

- Ausführliche Directoryausgabe
- Ausgabe der BAM
- Anzeige der möglichen Anfangsblöcke von Programmen
- (etwa falls die Disk ohne ID formatiert wurde)
- Überprüfung von Disketten und Laufwerk
- Gelöschte Dateien zurückholen
- Dateien/Disketten vor dem Löschen schützen
- Diskettenkommandos ausführen
- Anzeige von Dateien mit oder ohne Commodore-Steuerzeichen
- Änderung von Disknamen und ID

Tabelle 1. Die wichtigsten Fähigkeiten von XDOS

XDOS (Listing) arbeitet mit den meisten Floppy-Speedern problemlos zusammen und wird mit

```
LOAD "XDOS",8,1
```

geladen und dann entweder durch Betätigen von <RESTORE>, einen Reset (hardwaremäßig oder SYS 64738) oder mit

```
SYS (PEEK(32768)+256*PEEK(32769))
```

gestartet. Wenn Sie den SYS-Befehl benutzen, sollten Sie diese Adresse notieren.

Da noch 30 KByte des Basic-Speichers frei sind, kann man also gleichzeitig Basic- oder Maschinenprogramme schreiben beziehungsweise im Speicher halten (bei Maschinenprogrammen muß allerdings der Speicherbelegungsplan dieses Programmes (Tabelle 2) beachtet werden). Diese Platzersparnis verdeutlicht auch, warum der Menübildschirm nicht im Grafikmodus programmiert wurde; dies wäre zwar vom Optischen her anspruchsvoller gewesen, hätte aber auch zirka 10 KByte Speicher verbraucht und obendrein zu einer unnötigen Verlangsamung des Programmlaufes geführt.

Außerdem sind die 4 KByte im Adreßbereich 49152-53247 ($C000-$CFFF) hinter dem Basic-ROM frei, so daß dort ein Maschinensprache-Monitor oder ähnliche Utilities abgelegt werden können.

Integrierte Centronics-Schnittstelle

Für die Besitzer von Druckern am Parallelport wurde eine softwaremäßige Centronics-Schnittstelle integriert, so daß Sie einen Drucker mit einem geeigneten Kabel am Userport anschließen und betreiben können. Es kann vorkommen, daß vor dem ersten Ausdruck der Drucker auf OFF-LINE und dann wieder auf ON-LINE geschaltet werden muß, um eine korrekte Initialisierung zu erreichen.

Allgemeine Hinweise

Bei gleichzeitiger Benutzung eines Diskmonitors müssen Sie beachten, daß dieser normalerweise hexadezimale Parameter bei der Angabe von Spur und Sektor verlangt, während XDOS Dezimalzahlen ausgibt. Vor Verwendung seiner Daten müssen Sie diese Zahlen mittels des »#«-Befehls der meisten Monitore in Hexadezimalziffern umwandeln.

Nach dem Start durch einen Reset wartet das Programm, bis es anhand des Zustands der seriellen Datenübertragungsleitung zum Diskettenlaufwerk feststellt, daß dieses seinerseits den Reset beendet hat.

Danach werden die Systeminformationen in der Datei »XDos Pr/Dsk data« gespeichert und anschließend in den Speicher geladen. Der Directory-Header und der erste Directory-Sektor werden ebenfalls ermittelt und in den Speicher übertragen.

Daraufhin erscheint der Menübildschirm. Auf ihm sind links die einzelnen Menüpunkte aufgeführt, die beim dortigen Positionieren des Mauszeigers invertiert und durch anschließendes Betätigen des Feuerknopfes aufgerufen werden. Das geschieht allerdings erst nach Loslassen des Feuerknopfes, womit Probleme durch zu langes Warten bei mit »*« gekennzeichneten Funktionen (siehe unten) vermieden werden. Auf der rechten Seite des Bildschirms sind im obersten Abschnitt der Diskettenname, darunter die einzelnen Dateinamen mit dem jeweiligen Dateityp aufgeführt. Unter diesen werden die speziellen Informationen einer Datei aufgelistet. Der letzten Bildschirmzeile sind die wichtigsten Systemparameter zu entnehmen.

Auch Tastensteuerung möglich

Die Steuerung des Pfeiles kann alternativ durch Kombination <CTRL> und einer der folgenden Tasten erfolgen:

<CRSR links/rechts>:	oben
<A>	unten
<D>	links
<G>	rechts
<J>	Feuer

Bei vielen der im folgenden beschriebenen Funktionen ist eine Druckerausgabe vorgesehen. Falls der Drucker ausgeschaltet ist, wird keine Fehlermeldung ausgegeben und der ausgewählte Menüpunkt ohne Druckerausgabe abgearbeitet (zumindest bei am Parallelport angeschlossenen Geräten; um diese Funktion zu nutzen, sollten Sie im Printer-Menü auf jeden Fall das Interface »Centronics« wählen und den Drucker mindestens einmal nach dem Computer angeschaltet haben). Dies ist beispielsweise bei CHECK DISK nützlich, wenn man sich nur allgemein über den Zustand einer Diskette informieren möchte.

Bitte schalten Sie den Drucker nach dem Computer ein, da sonst (hardwarebedingt) ein benötigtes Bit im CIA 2 nicht gesetzt wird und damit keine Datenübertragung an den Drucker möglich ist.

Während der Datenübertragung zur Diskettenstation werden die Sprites für den Mauszeiger abgeschaltet. Ansonsten könnte es zu Timing-Problemen kommen, da die Sprites während des NMI des VIC erzeugt und gesteuert werden.

Falls Fehler bei Diskettenzugriffen auftreten, werden diese in der untersten Bildschirmzeile auf der rechten Seite angezeigt und es erscheint eine Meldung (im weiteren »Disk-Error Window« genannt). Danach kann durch Betätigung des Feuerknopfes fortgefahren werden.

Andere Fehler bei Gerätezugriffen werden durch die Betriebssystemmeldung »I/O Error #X« angezeigt. Die Bedeutung der Fehlernummern (»X«) im einzelnen:

Wert von »X«	Bedeutung
0	Abbruch durch STOP-Taste
1	Too many files
2	File open
3	File not open
4	File not found
5	Device not present
6	Not input file
7	Not output file
8	Missing Filename
9	Illegal device number
240	RS 232 Channel Open/Close

In der Praxis dürften, wenn überhaupt, nur die Fehler 0, 4 und 5 vorkommen. Bei diesen Fehlern genügt es meistens die RESTORE-Taste zu drücken oder einen Reset auszulösen, um das Programm wieder neu zu starten.

Das Programm achtet bei einigen Routinen (beispielsweise NEXT) auf die Übereinstimmung der Daten des »Di-

Adresse	Belegung
$fb,$fc	*DSKPNT (Zeiger auf aktuellen Diskbuffer)
$fd,$fe	Temporäre Zeiger
$0400-$07ff	1 KByte Bildschirmspeicher
$0801-$7fff	30 KByte Basic-Programmspeicher
$8000-$bfff	16 KByte XDOS (RAM)
$8000-$8001	Kaltstartvektor
$8002-$8003	Warmstartvektor
$8004-$8008	CBM80 Autostart-Identifikation
$8009-$800a	*Sprungvektor für User- Interface-Init
$800b-$800c	*Sprungvektor User-Printer-Init
$800d-$800e	*Sprungvektor User-Druckfilter
$800f-$8010	*Sprungvektor User- Druckausgabe

Der Stern (»*«) vor der Erklärung weist darauf hin, daß diese Speicherstelle beim Programmieren eigener Routinen nützlich ist bzw. beachtet werden muß.

Tabelle 2. So wird der Speicher des C64 von XDOS belegt

rectory Headers« im Speicher, mit denen auf der im Moment eingelegten Diskette. Wenn Sie es vorher festgelegt haben (erkennbar an »←« hinter der Disk-ID, einstellbar mittels »Diskwechsel anzeigen«, siehe dort), werden Sie auf diesen Diskwechsel aufmerksam gemacht, indem eine Fehlermeldung und der Name der eingelegten Disk ausgegeben wird. Der Name und die ID der Diskette im Speicher ist ja auf dem Bildschirm zu sehen. Sie haben nun die Möglichkeit, diese Diskrepanz zu übergehen (»Ignore«) oder die neuen Daten einzulesen (mittels »INIT«, das wie NEW.DSK arbeitet).

Die Routine INIT macht immer den ersten Directory-Sektor zum aktuellen, im rechten Bildschirmabschnitt sichtbaren Sektor.

Bei allen mit einem Stern (»*«) gekennzeichneten Funktionen erscheint eine Anfrage (nennen wir sie im weiteren Standard-Window), ob diese Funktion ausgeführt (»OK«) oder abgebrochen (»Cancel«) werden soll, was durch Anklicken entschieden wird.

Die mit einem Punkt versehenen Funktionen erwarten eine Tastatureingabe; sie werden durch Betätigung der Return-Taste oder des Feuerknopfes abgebrochen.

Weitere Funktionen

Alle Funktionen, bei denen eine Druckausgabe vorgesehen ist, geben zuerst die Überschrift und dann eine Zeile mit allgemeinen Informationen über die aktuelle Diskette aus. Die Überschrift enthält den Disknamen, die ID und die Bezeichnung des Aufzeichnungsformates (Vic 1541, kompatibel mit CBM 2031/4040, oder CBM 8050/8250). Nun zu den einzelnen Funktionen des Hauptmenüs:



Bild 1. Das Auswahlfenster für »XDirectory«, »Rescratch« und »Scratch Protect«

XDirectory (eXtended Directory) – Ausgabe aller Datei-Informationen auf den Drucker

In einem sehr umfangreichen Fenster, das nach Anklicken dieses Punktes erscheint (Bild 1), sind nahezu alle Charakteristika der gewünschten Dateien festlegbar. Diese sind im einzelnen:

Type(s):	Durch Anklicken eines Typs (Prg, Seq, Rel, User oder DEL) wählen Sie einen oder mehrere Dateitypen aus; die von Ihnen bestimmten Parameter werden im Menü durch ein Häkchen gekennzeichnet.
Attributes: WFO:	»Write File Open«; nicht geschlossene Dateien (kenntlich an einem Stern (*) beim Auflisten des Directory).
Closed:	ordnungsgemäß geschlossene Dateien
R/W:	ungeschützte Dateien
Protected:	mit Löschschutz versehene Dateien (im Directory mit einer spitzen Klammer (»<«) gekennzeichnet).
Name:	Der bis zu 16 Zeichen lange Dateiname darf Joker (* und ?) enthalten; diese sind genau wie bei normalen DOS-Befehlen zu benutzen (siehe VC-1541 Handbuch).

Falls alle Dateien für das erweiterte Directory verwendet werden sollen, müssen alle Menüpunkte angekreuzt sein und als Dateiname nichts oder ein Stern (»*«) eingegeben werden.

XDOS achtet darauf, daß in jeder Zeile mindestens ein Punkt angekreuzt ist; wenn Sie also bei den Dateitypen das letzte vorhandene Häkchen löschen wollen, müssen Sie zuerst einen anderen Dateityp auswählen. In den Attribut-Zeilen wird durch Anklicken des letzten angekreuzten Menüpunktes dieser gelöscht und der jeweils andere aktiviert.

Nach Bestätigung durch <RETURN> beziehungsweise Anwählen von »OK« werden die erweiterten Informationen der Dateien, die Ihrer Auswahl entsprechen, auf dem Drucker ausgegeben. In den Schriftarten Pica beziehungsweise Elite (anzuwählen im Printer-Menü), werden in der Überschrift die Abkürzungen von »Anzeige der Datei-Informationen« verwendet, bei »Compressed« die ausgeschriebenen Begriffe.

Abschließend werden die freien (left) und belegten (used) Blöcke der Diskette ausgegeben (einschließlich freier Directory-Blöcke).

BAM – Ausgabe der Block Availability Map auf Drucker

Es werden untereinander die einzelnen Tracks und vertikal deren jeweilige Sektoren ausgegeben, wobei ein Stern (»*«) einen belegten, ein Punkt (».«) einen freien und ein Leerzeichen einen nicht vorhandenen Block kennzeichnen. Des weiteren werden die vorhandenen (available), benutzten (used) und freien (free) Blöcke pro Spur am Ende jeder Zeile ausgedruckt.

Abschließend wird die gesamte Diskettenkapazität in Blöcken (Total disk capacity in blocks), die benutzten beziehungsweise belegten (Blocks used/allocated) und freien (Blocks free/left capacity) ausgedruckt.

Diese Angaben beziehen sich auch auf die Directory-Blöcke, also können andere Werte als bei »xxx blocks free.« der normalen Directory-Anzeige-Funktion ausgegeben werden.

Vor der Rückkehr zum Hauptmenü wird der Directory-Header (Track 18, Sektor 0) und der erste Directory-Block (18,1) eingelesen. Dies wird auch bei den Funktionen »Rescratch, Scratch, Protect und Disk Menu« durchgeführt. Sie brauchen also keine Angst zu haben, es gehen keine Daten verloren.

Hidden Files - Liste möglicher Anfangsblöcke von Dateien

Alle Blöcke, die erste Datenblöcke einer Datei sein könnten, und danach die Anzahl der möglichen Dateien (xxx potential files) werden auf dem Drucker (abhängig von der Schriftart, zwei- oder dreispaltig) ausgegeben. Mittels dieser Liste ist es möglich, den Directory-Eintrag eines völlig aus dem Directory gelöschten Programmes (beispielsweise durch Formatieren ohne ID, fehlerhaftes Beschreiben eines Directory-Sektors) mit einem Diskmonitor wiederherzustellen.

Disketten überprüfen

Check Disk - Überprüfung einer Diskette auf Lesefehler

Die eingelegte Diskette wird auf eventuelle Beschädigungen hin überprüft und aufgetretene Fehler auf dem Drucker (je nach Schriftart in zwei oder drei Spalten) angezeigt. Bei jedem aufgetretenen Lesefehler erscheint das »Disk-Error-Window«, in dem ein Fehler durch Klicken mit Joystick oder Maus bestätigt werden muß.

Wenn kein Fehler entdeckt wurde, wird die Meldung »Disk successfully read.« ausgegeben, ansonsten »xxx Errors detected.«.

Performance Test - Überprüfung des Disklaufwerkes

Dieser Programmteil arbeitet ähnlich dem Performance Test-Programm der Test/Demo-Diskette; zuerst wird die eingelegte Diskette neu formatiert, das heißt alle darauf enthaltenen Daten werden gelöscht! Nehmen Sie also nur eine Diskette, die entweder unformatiert ist oder keine wichtigen Daten enthält. Nacheinander wird Sektor 0 der Spuren 18, 1 und 35 beschrieben und danach wieder gelesen. Falls irgendwelche Fehler auftreten, könnte dies an einem dejustierten Lesekopf liegen. Falls der Formatiervorgang unterbrochen wurde, beziehungsweise fehlerhaft war, sollte ein Reset ausgelöst werden (wodurch der C64 ja sofort wieder in XDOS springt).

Gelöschte Dateien holen

Rescratch - Wiederherstellung gelöschter Dateien

Hiermit holt man gelöschte Dateien zurück, sofern kein weiterer Schreibzugriff auf die Diskette nach dem Löschen erfolgte.

Es erscheint das gleiche Auswahlfenster wie beim Menüpunkt »XDirectory«, mit dem Unterschied, daß hier nur der Dateiname zur Erkennung der gewünschten Datei(en) verwendet wird (da beim Löschen einer Datei der Status auf WFO und R/W gesetzt wird, sind diese vom Programm angekreuzt); Sie dürfen immer nur einen Dateityp auswählen, außerdem wird das Ankreuzen des DEL-Typs unterbunden. Nach Ausführung des Rescratch-Befehls wird ein Validate ausgeführt, um die Blöcke der Datei(en) in der BAM zu belegen. Falls hierauf die Fehlermeldung »Illegal Track or Sector« angezeigt wird (und natürlich nur, wenn die anderen Dateien der Disk intakt sind!), ist dies ein sicherer Hinweis darauf, daß bereits Schreibzugriffe auf diese Bereiche der Diskette erfolgten und damit eine oder mehrere Dateien nicht mehr zurückzuholen sind. Sie sollten diese Datei(en) wieder aus dem Inhaltsverzeichnis löschen und möglichst ein Validate ausführen (im DISK MENU möglich).

Scratch Protect — Löschschutz für Dateien

Dieser Programmpunkt erlaubt das Schützen beziehungsweise Freigeben einer Datei vor dem Löschen (mit dem Scratch-Befehl, kein Verhindern des Formatierens!).

Das Auswahlfenster arbeitet wie bei Rescratch, es dürfen jedoch mehrere Dateitypen angekreuzt werden. Zusätzlich wird durch Anklicken von R/W beziehungsweise Protected das Freigeben beziehungsweise Schützen von Dateien ausgewählt. Es läßt sich hierbei immer nur ein Modus aktivieren! Darüber hinaus werden Dateien, die zwar Ihrer Auswahl entsprechen, den gewünschten Status jedoch schon besitzen, nicht von neuem präpariert; hierdurch wird die Ausführungsgeschwindigkeit erhöht.

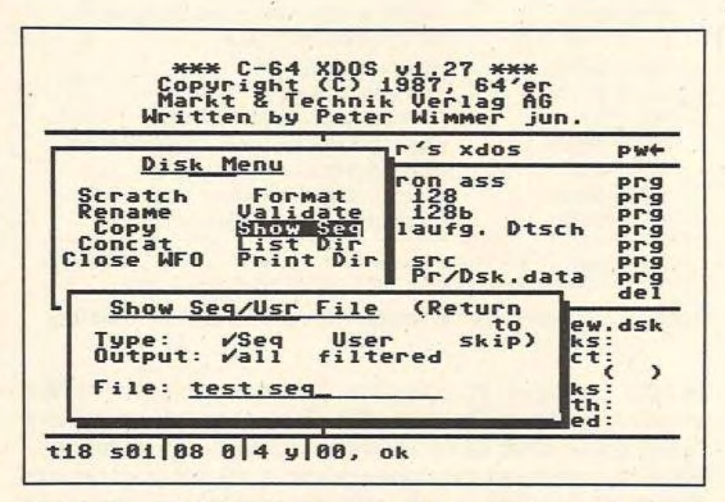


Bild 2. Das DISK MENU-Fenster, mit dem Untermenü »Show Seq« — so übersichtlich können C 64-Programme sein

Disk Write Protect - Schreibschutz für gesamte Disk

Hiermit läßt sich eine gesamte Diskette vor Schreibzugriffen schützen, was auch DOS-Befehle wie Scratch, Validate etc. miteinbezieht. Solchermaßen geschützte Disketten erkennt man in der Überschriftzeile am Aufzeichnungsformat »CBM 8050/8250«.

Disk Address Change - Softwaremäßiges Umstellen der Geräteadresse des Laufwerks

Nachdem man alle anderen eventuell vorhandenen Laufwerke ausgeschaltet hat, bestimmt man durch Anklicken die alte (jetzige) und neue Gerätenummer. Der gewählte Wert erscheint in dem kleinen Window rechts oben. Anschließend wird die Geräteadresse des Laufwerks softwaremäßig geändert. Dies bedeutet, daß nach einem Reset oder nach Ausschalten dieser Diskstation diese Einstellung verloren ist. Darüber hinaus wird die aktuelle Gerätenummer im XDOS für Diskzugriffe gleich auf die neue Nummer geändert.

Es kann vorkommen, daß diese Routine mit manchen Hardware-Floppy-Speedern nicht läuft, was sich meist in einem Absturz äußert (Reset auslösen!). Mit dem Original-C 64-Betriebssystem funktioniert sie jedoch einwandfrei.

DISK MENU

In diesem Menüpunkt sind mehrere Funktionen zu einem Untermenü (Bild 2) zusammengefaßt (wichtig für Erweiterungen anderer Funktionen).

Falls die Verwendung von Jokern (? und *) nicht durch Angabe in den folgenden Überschriften erlaubt ist, führt sie immer zu Fehlermeldungen.

Scratch - Löschen von Dateien (? *)

Hiermit werden Dateien der aktuellen Disk gelöscht.

Rename - Umbenennen einer Datei

Der aktuelle Name wird hinter »File« eingetragen, der neue bei »New Name«.

Copy – Kopieren/Zusammenfassen von Dateien unter neuem Namen

Kopieren: »File« ist der aktuelle Name, »New Name« der Name der Kopie.

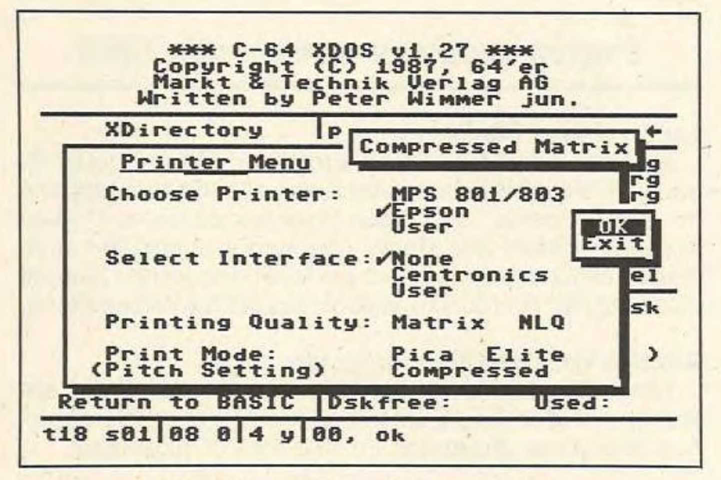


Bild 3. Das Printer Menu (Einstellung der Drucker-Parameter) zeigt die Anpassungsfähigkeit von XDOS.

Zusammenfassen: »File« sind die zusammenzufassenden Dateien, »New Name« der Name der neu erzeugten Datei.

Concat – Anhängen einer Datei an eine andere

Diese Funktion dient zum Anhängen der Daten der Datei »File« an die Datei »New Name«. New Name bedeutet hier also nicht, daß eine neue Datei erzeugt wird.

Close WFO (Write File Open) – Schließen einer nicht korrekt geschlossenen Datei

Wenn man eine sequentielle Datei zum Schreiben öffnet und dann versäumt, sie wieder zu schließen (beispielsweise bei einem Stromausfall), erhält man beim Leseversuch dieser Datei die Fehlermeldung »Write File Open«. Zur Wiederherstellung solcher Dateien dient diese Funktion, wobei im erscheinenden Fenster »File« die WFO-Datei ist, und »New Name« für die restaurierte, neue Datei steht.

Format – Formatieren/Löschen von Disketten

Formatieren: Hierzu muß sowohl der Diskname als auch die ID angegeben werden.

Achtung! Alle Daten auf der zu formatierenden Disk werden unwiederbringlich zerstört!

Löschen: Bei alleiniger Angabe des gewünschten Disknamens (also ohne ID) wird nur der erste Directory-Sector (Track 18, Sector 0) gelöscht; siehe Hidden-Files.

Validate – Freigabe unbenutzter Blöcke

Diese Funktion gibt alle Blöcke, die nicht von Dateien belegt sind, wieder frei. Falls das Validieren während der Ausführung abgebrochen wird, besteht die Möglichkeit, daß der Directory-Header mit der BAM nicht korrekt gespeichert und somit die Diskette unbrauchbar wird.

Außerdem kann es bei Disketten, auf denen absichtlich Blöcke als belegt gekennzeichnet wurden (beispielsweise um die weitere Benutzung eines zerstörten Blocks zu verhindern), zu Problemen kommen.

Show Seq/Usr File – Anzeigen einer sequentiellen beziehungsweise User-Datei auf dem Bildschirm

Die Daten einer sequentiellen oder User-Datei (dies legt man durch Anklicken des entsprechenden Wortes bei »Type« fest) werden entweder ohne Filter (all) oder mit den Steuercodes als Commodore-Grafikzeichen auf dem Bildschirm ausgegeben. Dies kann zum Lesen von in Dateiform vorhandenen Anleitungen oder zur Analyse fremder Dateistrukturen genutzt werden.

List Directory – Auflisten des Diskinhalts am Bildschirm (? *)

Hiermit läßt man entweder alle Dateien (»List all files«), nur die eines bestimmten Types (»Prg«/»Seq«/»Usr«/»Rel«) oder die Dateien (aller Typen), die mit dem eingegebenen Namen übereinstimmen, auf den Bildschirm ausgeben. Nachdem das Window voll ist, wird durch Anklicken von »Cont« die nächste Seite angezeigt oder durch »Abort« die Ausgabe abgebrochen.

Print Directory – Ausgabe des Directory auf Drucker (?*)

Das Directory wird hier entsprechend der Anwahl des Filetyps auf dem Drucker ausgegeben.

Nach Beendigung der im Disk-Menü angegebenen Routinen wird zum DISK MENU zurückgekehrt und der Disk-Status angezeigt.

Return – Zurück zum Hauptmenü

Das DISK MENU-Window verschwindet und man befindet sich wieder im Hauptmenü.

Printer Menu – Einstellung der Drucker-Parameter

Hier legen Sie die Konventionen für die Datenübertragung an den Drucker fest (Bild 3). Besonders wichtig ist dabei die richtige Verwendung des Punktes »Interface«. Die drei Wahlmöglichkeiten haben folgende Bedeutung:

Druckeransteuerung

None	normale Übertragung über seriellen IEC-Bus, bei Commodore-Druckern oder Druckern mit Interface
Centronics:	parallele Datenübergabe für Drucker mit Centronics-Schnittstelle, beispielsweise Epson. Hier ist die Sekundäradresse belanglos, da die Druckerausgabe über eine eigene Routine läuft!
User	selbstprogrammierte Übergaberoutine, beispielsweise RS232C

Print Mode (Pitch Setting):

Hier wird der Druckmodus und die Zeichenbreite gewählt. Folgende Schrifttypen stehen – natürlich nur bei Druckern, die diese auch beherrschen – zur Auswahl:

Pica (10 cpi), Elite (12 cpi), Compressed (17 cpi). Die Einheit »cpi« bedeutet »characters per inch«, also die Anzahl Zeichen pro 2,54 Zentimeter.

Der gewählte Drucker und das Interface werden durch ein Häkchen markiert, die Druckqualität und Zeichenbreite werden in dem kleinen Window rechts oben angegeben.

Bei Auswahl eines Commodore-Druckers wird die Schriftart automatisch auf Pica gesetzt, da nur 80 Zeichen/Zeile verfügbar sind; allerdings kann dies, falls erforderlich, nachträglich wieder geändert werden.

Falls ein Drucker mit Centronics-Schnittstelle an dem seriellen Port mittels eines Interfaces angeschlossen ist, muß natürlich als Interface »None« gewählt werden.

Bei Anklicken von »OK« wird die aktuelle Einstellung auf Disk gespeichert und bei jedem Neustart des Programms eingelesen, bei Anwahl des Menüpunktes »Exit« merkt das Programm sich die Einstellung nur bis zum nächsten Start.

System Settings – Einstellung der Systemparameter

Diese Parameter lassen sich sowohl durch Aufruf dieses Menüs als auch durch Anklicken der Anzeige in der untersten Bildschirmzeile (schrittweise) ändern. Folgende Parameter sind in diesem Menü einstellbar:

Disk:
Drive No: Entweder Laufwerk 0 oder 1. (Standard: 0)
Device No: Es lassen sich vier verschiedene Diskstationen mit den Gerätenummern von 8 bis 11 ansprechen. (Standard: 8)
Announce Disk Exchange:
Um das Auswechseln der Disk im aktuellen Laufwerk anzuzeigen, muß hier »Yes« angeklickt werden. (Diese Funktion ist nützlich, um Fehlbedienungen vorzubeugen.)
Printer:
Auto Line/Feed: Zum Ein- beziehungsweise Ausschalten der in diesem Programm integrierten Linefeed-Funktion.
Device No: Gerätenummer des Druckers, 4 oder 5. (Standard: 4)
Secundary Addr: Durch Anklicken der Zeichen »>« oder »<« wird die Sekundäradresse erniedrigt beziehungsweise erhöht. Dies dient zur Ausnutzung besonderer Drucker- beziehungsweise Interface-Eigenschaften.

Wenn Sie das »OK«-Feld anklicken, wird die aktuelle Einstellung auf Disk gespeichert, bei »Exit« lediglich bis zum nächsten Start des Programmes verwendet (s.o.).

Quit - Verlassen des Programms mit Reset
Das Programm wird verlassen, die Autostart-Funktion abgeschaltet und ein Reset ausgeführt. Von diesem Zeitpunkt an muß das Programm, um es erneut zu verwenden, wieder von Disk geladen werden, da es durch eine CBM-Kernel Routine, die nach einem Reset einen Speichertest durchführt, teilweise zerstört wird. Danach stehen die oben angegebenen Startmöglichkeiten zur Verfügung.

Break - Sprung in einen eventuell installierten Maschinensprachemonitor
Es wird der »Break«-Befehl des 6502/6510-Prozessors (BRK; Code: $00) ausgeführt und, falls ein solcher installiert ist, in einen Maschinensprachemonitor verzweigt (sonst geschieht das gleiche wie nach einem NMI).

Aus diesem erfolgt der Rücksprung ins XDOS mit Eingabe von <G> und Drücken der Return-Taste (natürlich nur, wenn der Programm-Counter (PC) nicht verändert wurde!).

Return to Basic
Man gelangt zurück ins CBM V2 Basic, mit all den oben genannten Möglichkeiten des Neustarts, also Drücken von <RESTORE>, Reset oder SYS-Befehl.

Änderung des Disknamens und der ID
Mit dieser Funktion ändern Sie den Namen und die ID (die in der ersten Zeile bei der Directory-Funktion angezeigt werden). Den aktuellen Diskettennamen und die ID sehen Sie noch auf dem Bildschirm.

Diskwechsel anzeigen
Wenn Sie das ID-Feld anklicken, erscheint beziehungsweise verschwindet abwechselnd ein »←« hinter der ID. Hiermit erlauben oder unterbinden Sie das Anzeigen eines Diskwechsels.

Achtung: Um sicherzugehen, daß Sie etwaige Manipulationen immer auf der richtigen Diskette ausführen, sollten Sie diese Funktion immer angeschaltet lassen, auch wenn sich die Zugriffsgeschwindigkeit etwas reduziert!

Im rechten unteren Kasten des Hauptmenüs befinden sich folgende Punkte:

Previous
Der vorhergehende Directory-Block wird gelesen und am Bildschirm angezeigt; falls der aktuelle Block der erste Block ist (Spur 18, Sektor 0), wird das Anklicken ignoriert.

Next
Der nächste Directory-Block wird eingelesen und angezeigt.

New.dsk
Sowohl der Directory-Header (Track 18, Sektor 0) als auch der erste Directory-Block wird eingelesen; hiermit aktualisieren Sie nach einem Diskwechsel die Daten im Computer.

Starten von Programmen aus XDOS

Laden eines PRG-Files
Sie haben die Möglichkeit, Programm-Dateien (erkennbar am Filetyp PRG) durch Anklicken absolut (das heißt wie mit LOAD "name",8,1) in den Speicher zu laden. Es wird der NEW-Befehl des Basic-Interpreters ausgeführt und, falls es sich bei der Datei um ein Basic-Programm handelt (Start mit RUN bei $0801), wird dieses auch sofort gestartet.

Anzeige der Datei-Informationen
Wenn Sie den Datei-Typ einer bestimmten Datei anklicken, werden im darunterliegenden Fenster Informationen über diese angezeigt. Es sind dies im einzelnen:

Adr:	Startadresse (nur bei PRG-Dateien)
Blocks:	durch diese Datei belegte Blöcke
Closed:	Y bei geschlossenen Dateien, N bei WFO (Datei nicht geschlossen). Durch Anklicken zu verändern.
Protect:	Y bei geschützten, N bei ungeschützten Dateien. Durch Anklicken zu verändern.
1st T: ( )	Spur des ersten Datenblocks (bei REL-Dateien: Spur des ersten Side-Sektor-Blocks)
S: ( )	wie oben, jedoch Sektor
Side Sector	Anzahl der Side-Sektor-Blocks (nur Blks: bei RELativen Dateien)
Recs:	Anzahl Records in einer REL-Datei
Length:	Recordlänge (nur bei REL-Dateien)
Dskfree:	freie Blöcke dieser Disk
Used:	belegte Blöcke

Die Zahl der Records einer relativen Datei kann etwas höher liegen, als Sie es festgelegt haben; dies liegt jedoch nicht am Programm (die sehr komplizierte Berechnung stimmt genau), sondern an dem Umstand, daß das DOS der VC-1541 den letzten Datenblock einer REL-Datei voll ausnutzt. Falls also durch Ihre Record-Anzahl ein Block nicht ganz belegt wird, gibt das DOS die restlichen möglichen Records dieses Blocks gleich mit frei.

Bei der Anzeige der freien Blocks auf der Diskette werden auch die freien Blöcke der Directory-Spur mitgerechnet.

Hinweise zur Programmierung von XDOS

XDOS liegt im Speicher ab 32768 ($8000, auch als Autostart-Bereich bekannt), und reicht mit seiner Länge von 16 KByte bis unter das Basic-ROM. Aus diesem Grund wird dieser Bereich auf RAM umgeschaltet, und nur bei Zugriffen auf Basic-ROM-Routinen (und natürlich beim Verlassen des Programms) der ursprüngliche Zustand wiederhergestellt. Daher müssen die Programmteile, die sich dieser Routinen bedienen, im RAM vor 40960 ($A000) liegen. Außerdem wird der Basic-Speicher um 8 K auf 32767 ($7fff) herabgesetzt, um ein Überschreiben des Programms zu verhindern.

Programmierung des Disk-Schreibschutzes
Das Formatkennzeichen in Byte $02 im Directory-Header (Spur 18, Sektor 0) wird von »A« auf »C« geändert, wodurch das DOS die Diskette zwar lesen, aber nicht beschreiben kann. Da die Wiederherstellung des ursprünglichen Zustandes eben wegen des Schreibschutzes nicht

möglich ist, wird durch das Beschreiben der Adresse $0101 (257) im Floppy-RAM mit dem ASCII-Wert des Buchstabens »A« eine VC-1541-Diskette simuliert (was ja im Endeffekt auch zutrifft).

Auf der Programmservice-Diskette befindet sich das Sourcecode-Listing des Programmes sowie als Textfile eine Information, wie die Routinen von XDOS auch von Ihren Programmen aus verwendet werden können. Damit haben Sie nun ein umfassendes Werkzeug zur Bedienung Ihrer Floppy-Station in der Hand und können auf viele Einzelprogramme mit gleichen Funktionen verzichten.

(Peter Wimmer jun./Andreas Lietz)

```
Name : xdos                    8000 b478
-----------------------------------------
8000 : 25 81 44 81 c3 c2 cd 38   21
8008 : 30 ad 81 ad 81 ad 81 ad   0c
8010 : 81 00 00 00 00 00 00 00   92
8018 : 00 00 00 01 02 03 04 07   8f
8020 : 40 00 00 80 00 00 00 00   71
8028 : 00 00 00 00 00 08 01 01   6f
8030 : 00 50 00 04 07 00 00 00   49
8038 : 00 00 00 00 00 00 00 00   39
8040 : 00 00 00 00 00 ea ea ea   19
8048 : ea ea ea ea ea ea ea ea   47
8050 : 4c 96 81 4c 02 82 4c 0b   4d
8058 : 82 4c 33 82 4c 3e 82 4c   77
8060 : 45 82 4c 86 82 4c 64 84   ef
8068 : 4c 79 84 4c 8f 84 4c 9f   a9
8070 : 84 4c 0b 85 4c 69 94 4c   89
8078 : 6f 94 4c 8e 94 4c 9e 94   66
8080 : 4c a8 94 4c b2 94 4c b8   41
8088 : 94 4c bc 94 4c 00 95 4c   b8
8090 : 27 95 4c 6e 95 4c 73 95   17
8098 : 4c 78 95 4c 5a 97 4c 6f   82
80a0 : 97 4c 96 98 4c 99 98 4c   a3
80a8 : b1 98 4c b4 98 4c f7 98   4c
80b0 : 4c fa 98 4c ff 98 4c 02   23
80b8 : 99 4c 07 99 4c 40 99 4c   32
80c0 : dc 99 4c 61 9a 4c 88 9a   0b
80c8 : 4c 91 9a 4c ee 9a 4c f9   f6
80d0 : 9a 4c 07 9b 4c 20 9b 4c   92
80d8 : 32 9b 4c 38 9b 4c 45 9b   5a
80e0 : 4c 4b 9b 4c 61 9b 4c 76   53
80e8 : 9b 4c 14 9c 4c 78 9c 4c   d6
80f0 : a9 9c 4c bd 9c 4c 16 9d   72
80f8 : 4c 5b 9d 4c 68 9d 4c 78   78
8100 : 9d 4c 8e 9d 4c a0 9d 4c   f3
8108 : 1b 9e 4c 3b 9e 4c ad 9e   2d
8110 : 4c 3c 9f 4c 4a 9f 4c 53   65
8118 : 9f 4c b2 a0 4c bb a0 ea   99
8120 : ea ea ea ea ea 20 a3 fd   d1
8128 : 20 15 fd 20 a9 ee b0 fb   23
8130 : 20 a9 ee 90 fb 58 20 53   9c
8138 : e4 20 bf e3 a2 fb 9a a9   60
8140 : 08 8d 2d 80 20 33 82 20   50
8148 : 8e 9d a9 40 20 90 ff 20   de
8150 : 45 82 20 0e 95 a9 00 a2   8c
8158 : 2c a0 80 20 d5 ff 20 07   e4
8160 : 99 20 40 99 a2 00 a0 80   fa
8168 : 86 37 84 38 a2 7d bd f0   a1
8170 : a4 9d 40 03 ca d0 f7 20   a7
8178 : 86 82 20 6b 82 20 50 82   24
8180 : 20 45 82 a2 64 a0 b7 a9   b5
8188 : 00 20 8e 94 ae 3e 05 e0   81
8190 : 20 d0 03 4c 29 82 20 55   35
8198 : 9e 20 76 9b 20 91 9a 20   91
81a0 : 1b 9e 20 b2 a0 bd ad 81   1a
81a8 : 48 bd ac 81 48 60 d9 82   1e
81b0 : 5f 83 93 91 27 93 13 94   6c
81b8 : ae 84 da 84 f9 85 16 85   1f
81c0 : 05 8c e6 8a 01 8a 0a 82   a8
81c8 : 01 82 28 82 85 86 9b 82   65
81d0 : 5f 87 62 87 65 87 68 87   c0
81d8 : 6b 87 6e 87 71 87 74 87   c8
81e0 : 22 82 db 86 de 86 e1 86   c2
81e8 : e4 86 e7 86 ea 86 ed 86   82
81f0 : f0 86 1c 82 c4 89 d1 89   6e
81f8 : a6 82 b8 82 c3 82 ce 82   ee
8200 : 12 82 20 8e 9d 20 3e 82   06
8208 : 4c 86 e3 a9 37 8d 08 80   c6
8210 : 6c fc ff 20 5b ff 20 3e   b1
8218 : 82 00 4c 44 81 20 07 9b   a2
8220 : 4c 96 81 20 20 9b 4c 96   59
8228 : 81 20 32 9b a9 01 8d 26   de
8230 : 80 d0 ea 48 a5 01 29 fe   e1
8238 : 09 06 85 01 68 60 48 a9   c4
8240 : 07 85 01 68 60 a9 ea 8d   71
8248 : 07 99 20 69 94 4c 79 94   0c
8250 : a2 2a 8e 78 b4 bd 04 a1   46
8258 : 9d 7a b4 bd 2e a1 9d ba   f3
8260 : b4 bd 58 a1 9d fa b4 ca   57
8268 : d0 eb 60 a9 58 a0 a2 20   d0
8270 : 5a 97 a9 40 a2 16 9d a9   ad
8278 : 07 9d 69 06 9d 01 05 9d   9a
8280 : b1 04 ca d0 f1 60 a2 00   ad
8288 : 8e 1b d0 8e 27 d0 a2 0d   47
```

```
8290 : 8e f8 07 e8 8e f9 07 e8   20
8298 : 8e 28 d0 60 ad 32 80 49   7b
82a0 : ff 8d 32 80 4c 96 81 ee   60
82a8 : 2d 80 ad 2d 80 c9 0c 90   ce
82b0 : 05 a9 08 8d 2d 80 4c 96   73
82b8 : 81 ad 2c 80 49 01 8d 2c   56
82c0 : 80 4c 96 81 ad 33 80 49   45
82c8 : 01 8d 33 80 4c 96 81 ad   48
82d0 : 35 80 49 0a 8d 35 80 4c   f6
82d8 : 96 81 a2 c6 a0 b0 a9 0a   fa
82e0 : 20 6e 95 f0 78 8a 48 20   f8
82e8 : 14 9c a2 4b a0 a6 a9 13   68
82f0 : 20 78 9c 20 32 9b 20 16   24
82f8 : 9d ae 31 80 e0 0f f0 09   a5
8300 : a2 5e a0 a6 a9 51 48 d0   b6
8308 : 07 a2 af a0 a6 a9 85 48   bf
8310 : 20 78 9c 68 aa ca 8e 8e   f9
8318 : b3 a9 2d 20 a9 9c ce 8e   c7
8320 : b3 d0 f6 20 5b 9d 20 3b   97
8328 : 9e 20 ee 9a 68 a8 88 88   e4
8330 : 20 7d 87 ac 31 80 c0 0f   be
8338 : f0 08 a2 ee a0 bd a9 50   f2
8340 : d0 06 a2 75 a0 bd a9 79   fc
8348 : 20 78 9c 20 5b 9d 20 61   b5
8350 : 9a 48 d0 d5 a2 34 a0 a7   9b
8358 : a9 27 20 78 9c 4c 6d 8c   a7
8360 : a2 5b a0 a7 20 27 95 20   9f
8368 : 14 9c a2 5b a0 a7 a9 17   fa
8370 : 20 78 9c 20 32 9b 20 16   a4
8378 : 9d a2 ff a0 a7 a9 92 20   cd
8380 : 78 9c a9 01 8d 2a 80 a9   50
8388 : 03 8d 7f b3 20 ee 9a ad   e7
8390 : 2a 80 20 f9 9a 8e 37 bc   39
8398 : 8d d8 bc ae 2a 80 20 78   ae
83a0 : 9d 98 20 f9 9a 8e 0d bd   9e
83a8 : 8d 0e bd 20 37 84 20 f9   bc
83b0 : 9a 8e 1b bd 8d 1c bd ad   1c
83b8 : 16 80 38 ed 7e b3 20 f9   d4
83c0 : 9a 8e 15 bd 8d 16 bd a0   60
83c8 : 00 a2 07 20 37 84 ad 7e   2b
83d0 : b3 3d ab b3 20 4b 84 c8   83
83d8 : cc 16 80 b0 05 ca 10 ee   aa
83e0 : 30 e7 a2 d5 a0 bc a9 4f   9c
83e8 : 20 78 9c 20 5b 9d ee 2a   22
83f0 : 80 ae 2a 80 e0 24 90 94   fd
83f8 : a9 20 a2 03 9d d6 a7 9d   25
8400 : fa a7 ca d0 f7 20 64 84   b6
8408 : 20 8f 84 a2 00 bd 00 01   55
8410 : f0 06 9d fa a7 e8 d0 f5   bb
8418 : 20 79 84 20 8f 84 a2 00   c2
8420 : bd 00 01 f0 06 9d d6 a7   34
8428 : e8 d0 f5 a2 93 a0 a7 a9   7a
8430 : 73 20 78 9c 4c 8f 85 ee   9a
8438 : 7f b3 98 48 ac 7f b3 b9   c9
8440 : d5 bb 8d 7e b3 68 a8 ad   a2
8448 : 7e b3 60 48 f0 03 a9 2e   eb
8450 : 2c a9 2a 8d 80 b3 98 48   26
8458 : 0a a8 ad 80 b3 99 dd bc   2b
8460 : 68 a8 68 60 a0 04 a9 00   13
8468 : aa 18 79 d5 bb 90 01 e8   4d
8470 : c8 c8 c8 c8 c0 90 d0 f1   9f
8478 : 60 a0 04 a9 ab a2 02 38   a7
8480 : f9 d5 bb b0 01 ca c8 c8   84
8488 : c8 c8 c0 90 d0 f1 60 86   22
8490 : 62 85 63 20 a3 84 20 3e   ed
8498 : 82 20 df bd 4c 33 82 86   4f
84a0 : 62 85 63 20 3e 82 a2 90   45
84a8 : 38 20 49 bc 4c 33 82 a9   96
84b0 : 00 8d 22 80 8d 23 80 8d   1f
84b8 : 21 80 8d 20 80 a2 e9 a0   87
84c0 : b0 a9 05 20 73 95 f0 10   52
84c8 : ad 25 80 09 80 9d d5 b9   09
84d0 : 20 0b 85 d0 f3 20 00 95   5d
84d8 : 4c 6d 8c a9 80 8d 23 80   35
84e0 : 8d 22 80 a9 08 8d 1f 80   3e
84e8 : ad 21 80 d0 03 8d 20 80   7e
84f0 : a2 09 a0 b1 a9 07 20 78   b9
84f8 : 95 f0 dd bd d5 b9 49 40   05
8500 : 9d d5 b9 20 0b 85 d0 f3   02
8508 : 4c 8f 85 ae 2a 80 ac 2b   03
8510 : 80 20 b4 98 4c 61 9a a2   60
8518 : 11 a0 24 8e 38 80 8c 39   80
8520 : 80 a2 07 a0 02 20 a0 9d   a6
```

```
8528 : a9 bc a0 a8 20 5a 97 a2   e5
8530 : 08 a0 24 8e 38 80 8c 39   8f
8538 : 80 a2 06 a0 1f 20 a0 9d   50
8540 : a9 a9 a0 a9 20 5a 97 a2   94
8548 : 0a 8e 78 b4 bd 82 a1 9d   00
8550 : 7a b4 bd 8c a1 9d ba b4   81
8558 : bd 96 a1 9d fa b4 ca d0   9e
8560 : eb ad 2d 80 20 f9 9a 8e   d7
8568 : 14 05 8d 15 05 d0 42 20   25
8570 : b2 a0 bd 7a 85 48 bd 79   b5
8578 : 85 48 60 94 85 97 85 9a   2c
8580 : 85 9d 85 ad 85 b0 85 b3   46
8588 : 85 b6 85 d2 85 8e 85 20   47
8590 : 3b 9e 4c 7d 81 a9 08 2c   bb
8598 : a9 09 2c a9 0a 2c a9 0b   c5
85a0 : 8d 2d 80 20 f9 9a 8e 14   bf
85a8 : 05 8d 15 05 d0 c1 a9 28   6c
85b0 : 2c a9 29 2c a9 2a 2c a9   71
85b8 : 2b 8d 26 b4 18 69 20 8d   32
85c0 : 27 b4 38 e9 40 8d ed 85   c0
85c8 : 20 f9 9a 8e 3c 05 8d 3d   fa
85d0 : 05 d0 9c a9 4c 8d 27 99   9b
85d8 : a2 20 a0 b4 a9 08 20 6f   83
85e0 : 94 20 07 99 20 69 94 a9   6c
85e8 : 20 8d 27 99 a9 7b 8d 2d   d3
85f0 : 80 20 07 99 20 40 99 4c   78
85f8 : 8f 85 20 32 9b a2 0e a0   e0
8600 : 1c 8e 38 80 8c 39 80 a2   5b
8608 : 08 a0 0a 20 a0 9d a9 03   8b
8610 : 8d 78 b4 a9 0a a2 13 a0   7f
8618 : 1d 20 9e 94 a9 0b 20 a8   44
8620 : 94 a9 0d ca a0 16 8d 7d   11
8628 : b4 8e bd b4 8c fd b4 20   f5
8630 : df 9d a9 c0 a0 a9 20 5a   ed
8638 : 97 a9 7a ae d7 bb e0 41   7a
8640 : d0 05 8d a2 05 f0 03 8d   49
8648 : ca 05 20 b2 a0 e0 06 f0   fe
8650 : 2c a9 43 e0 04 f0 02 a9   61
8658 : 41 8d d7 bb a9 4c 8d 27   4f
8660 : 99 a9 07 a2 19 a0 b4 20   8e
8668 : 6f 94 20 07 99 a9 20 8d   8d
8670 : 27 99 a9 00 20 6f 94 20   de
8678 : 45 9b 4c 2f 86 20 69 94   bc
8680 : 20 07 99 4c 8f 85 20 32   1e
8688 : 9b 20 76 9b a2 0e a0 24   aa
8690 : 8e 38 80 8c 39 80 a2 08   1e
8698 : a0 02 20 a0 9d a9 38 a0   9f
86a0 : aa 20 5a 97 20 3c 9f a9   9a
86a8 : 10 a2 65 a0 bc 20 ad 9e   37
86b0 : ae 12 80 f0 24 a9 a0 9d   f3
86b8 : 65 bc e8 e0 15 90 f8 a2   d0
86c0 : 0c a0 14 18 20 f0 ff a9   01
86c8 : 02 a2 77 a0 bc 20 ad 9e   ce
86d0 : 20 4a 9f 20 45 9b 20 07   c1
86d8 : 99 4c 8f 85 a2 00 2c a2   4c
86e0 : 20 2c a2 40 2c a2 60 2c   79
86e8 : a2 80 2c a2 a0 2c a2 c0   a1
86f0 : 2c a2 e0 bd d7 b9 29 07   5b
86f8 : c9 02 d0 dd 8a 48 20 8e   3b
8700 : 9d 68 aa a9 4c 20 d2 ff   c2
8708 : a9 cf 20 d2 ff a9 22 20   11
8710 : d2 ff ad 2c 80 09 30 20   24
8718 : d2 ff a9 3a a0 10 20 d2   4c
8720 : ff bd da b9 c9 a0 f0 04   59
8728 : e8 88 10 f2 a9 22 20 d2   89
8730 : ff a9 2c 8d 19 04 8d 16   d5
8738 : 04 ad 2d 80 20 f9 9a 8e   c8
8740 : 17 04 8d 18 04 a9 31 8d   2d
8748 : 1a 04 a2 05 86 c6 bd 28   93
8750 : b4 9d 77 02 ca 10 f7 20   3e
8758 : 3e 82 20 44 a6 4c 86 e3   17
8760 : a0 00 2c a0 20 2c a0 40   86
8768 : 2c a0 60 2c a0 80 2c a0   82
8770 : a0 2c a0 c0 2c a0 e0 20   f2
8778 : 7d 87 4c 99 81 98 48 20   3d
8780 : 55 9e 20 64 84 20 8f 84   4a
8788 : a2 00 bd 00 01 f0 09 9d   91
8790 : 8b 07 9d 57 a7 e8 d0 f2   dc
8798 : 20 79 84 20 8f 84 a2 00   42
87a0 : bd 00 01 f0 09 9d 95 07   9e
```

Listing. »XDOS« bringt alle Floppy-Funktionen unter einen Hut

```
87a8 : 9d 51 a7 e8 d0 f2 68 a8  8c
87b0 : 8c 29 80 a2 00 b9 da b9  f2
87b8 : 9d 75 bd 9d ee bd c8 e8  05
87c0 : e0 10 90 f1 ac 29 80 b9  94
87c8 : d7 b9 48 48 29 40 f0 03  f5
87d0 : a9 59 2c a9 4e 8d f5 06  9b
87d8 : 8d c2 bd 8d 29 be 68 29  64
87e0 : 80 f0 03 a9 59 2c a9 4e  09
87e8 : 8d ea 06 8d b9 bd 8d 22  22
87f0 : be 68 29 07 0a 0a a8 a2  e6
87f8 : 00 b9 2d b4 9d 87 bd 9d  ff
8800 : fe bd c8 e8 e0 04 90 f1  80
8808 : ac 29 80 b9 d8 b9 20 f9  70
8810 : 9a 8e 10 07 8d 11 07 8e  71
8818 : a4 bd 8d a5 bd 8e 12 be  c9
8820 : 8d 13 be b9 d9 b9 20 f9  fd
8828 : 9a 8e 19 07 8d 1a 07 8e  13
8830 : a7 bd 8d a8 bd 8e 15 be  50
8838 : 8d 16 be b9 f4 b9 aa b9  f2
8840 : f3 b9 20 8f 84 a2 00 bd  e3
8848 : 00 01 f0 0c 9d cd 06 9d  22
8850 : 90 bd 9d 04 be e8 d0 ef  fd
8858 : ac 29 80 b9 d7 b9 29 07  ee
8860 : c9 02 d0 2e 20 b2 94 ac  67
8868 : 29 80 b9 d8 b9 aa b9 d9  e6
8870 : b9 a8 20 99 98 ad d8 ba  88
8878 : aa ad d7 ba 20 8f 84 a2  1c
8880 : 00 bd 00 01 f0 0c 9d 9a  9a
8888 : bd 9d 0a be 9d bf 06 e8  30
8890 : d0 ef ac 29 80 b9 d7 b9  51
8898 : 29 04 d0 01 60 b9 ea b9  0b
88a0 : 20 f9 9a 8e 13 07 8d 14  fd
88a8 : 07 8e ac bd 8d ad bd 8e  33
88b0 : 19 be 8d 1a be b9 eb b9  ac
88b8 : 20 f9 9a 8e 1c 07 8d 1d  b8
88c0 : 07 8e af bd 8d b0 bd 8e  24
88c8 : 1c be 8d 1d be b9 ec b9  2b
88d0 : 8d 52 89 8d 8a 89 a2 00  1a
88d8 : 20 8f 84 a2 00 bd 00 01  25
88e0 : f0 0c 9d da bd 9d 35 be  b4
88e8 : 9d 6d 07 e8 d0 ef 20 b2  8d
88f0 : 94 ac 29 80 b9 ea b9 aa  64
88f8 : b9 eb b9 a8 20 99 98 a0  9d
8900 : 06 a2 0b bd d8 ba d0 05  82
8908 : ca ca 88 d0 f6 98 48 09  db
8910 : 30 8d 45 07 8d ed bd 8d  93
8918 : 3b be bd d9 ba a8 bd d8  f6
8920 : ba aa 20 99 98 68 a8 ad  35
8928 : d6 ba 38 e9 11 4a a2 00  95
8930 : 88 f0 08 18 69 78 90 f8  c4
8938 : e8 d0 f5 20 9f 84 20 3e  25
8940 : 82 20 0c bc a0 fe 20 a2  35
8948 : b3 a5 61 20 2b ba 20 0c  4b
8950 : bc a0 7b 20 a2 b3 a5 61  60
8958 : f0 65 20 12 bb 20 1b bc  e8
8960 : a2 06 b5 60 9d bf 89 ca  12
8968 : d0 f8 20 33 82 ae d6 ba  91
8970 : bd d5 ba a8 bd d4 ba aa  9e
8978 : 20 99 98 ac d6 ba 88 88  97
8980 : 20 3e 82 20 a2 b3 20 0c  c4
8988 : bc a0 7b 20 a2 b3 a5 61  98
8990 : f0 2d 20 12 bb a2 06 bd  c6
8998 : bf 89 95 68 ca d0 f8 86  b2
89a0 : 70 a5 61 20 6a b8 20 cc  c6
89a8 : bc 20 99 84 a2 00 bd 00  8c
89b0 : 01 f0 0c 9d 5f 07 9d cc  1e
89b8 : bd 9d 2e be e8 d0 ef 60  3d
89c0 : 00 00 00 00 20 f7 89     b4
89c8 : f0 2a b9 d7 b9 49 80 4c  b7
89d0 : dc 89 20 f7 89 f0 1d b9  80
89d8 : d7 b9 49 40 99 d7 b9 20  66
89e0 : b8 94 a2 12 ac 26 80 b9  3f
89e8 : 91 b3 a8 20 b4 98 ac 29  96
89f0 : 80 20 7d 87 4c 99 81 ad  c4
89f8 : ea 06 ac 29 80 29 7f c9  18
8a00 : 20 60 a2 11 a0 23 8e 38  e9
8a08 : 80 8c 39 80 a2 08 a0 02  1e
8a10 : 20 a0 9d a9 c4 a0 aa 20  59
8a18 : 5a 97 a2 09 a0 21 8e 38  c5
8a20 : 80 8c 39 80 a2 07 a0 1f  68
8a28 : 20 a0 9d a2 03 bd 1d ac  40
8a30 : 20 d2 ff e8 e0 10 d0 f5  94
8a38 : a2 10 8e 78 b4 bd c7 a1  31
8a40 : 9d 7a b4 bd d7 a1 9d ba  76
8a48 : b4 bd e7 a1 9d fa b4 ca  23
8a50 : d0 eb ad 34 80 a2 00 20  65
8a58 : 8f 84 a0 00 b9 00 01 f0  d3
8a60 : 06 99 99 06 c8 d0 f5 a9  98
8a68 : 20 c0 03 f0 06 99 99 06  67
8a70 : c8 d0 f6 20 91 9a 20 b2  36
8a78 : a0 bd 81 8a 48 bd 80 8a  32
8a80 : 48 60 c0 94 8e 85 a1 8a  6c
8a88 : a4 8a ab 8a ae 8a b1 8a  c9
8a90 : b4 8a bb 8a be 8a c5 8a  3a
8a98 : c9 8a d0 8a d3 8a e0 8a  56
8aa0 : da 8a a9 00 2c a9 01 8d  59
8aa8 : 2c 80 10 c7 a9 08 2c a9  f0
8ab0 : 09 2c a9 0a 2c a9 0b 8d  d2
8ab8 : 2d 80 10 b7 a9 ff 2c a9  bf
8ac0 : 00 8d 32 80 4c 73 8a a9  01
8ac8 : 00 2c a9 0a 8d 35 80 10  2f
8ad0 : a2 a9 04 2c a9 05 8d 33  2d
8ad8 : 80 10 98 ee 34 80 4c 52  81
8ae0 : 8a ce 34 80 4c 52 8a a2  b5
8ae8 : 15 a0 22 8e 38 80 8c 39  d4
8af0 : 80 a2 07 a0 02 20 a0 9d  76
8af8 : a2 0d a0 24 8e 38 80 8c  93
8b00 : 39 80 a2 0b a0 22 20 a0  60
8b08 : 9d a2 0d 8e 78 b4 bd a0  71
8b10 : a1 9d 7a b4 bd ad a1 9d  c0
8b18 : ba b4 bd ba a1 9d fa b4  4f
8b20 : ca d0 eb a2 07 a0 23 8e  c1
8b28 : 38 80 8c 39 80 ca a0 14  f4
8b30 : 20 a0 9d a9 aa a0 ab 20  db
8b38 : 5a 97 ae 2e 80 bd b5 8b  b3
8b40 : aa a9 7a 9d 7d 05 ae 2f  2a
8b48 : 80 bd b5 8b aa a9 7a 9d  a3
8b50 : 1d 06 20 df 9d ae 31 80  89
8b58 : e0 50 d0 08 a2 a4 a0 ac  c1
8b60 : a9 04 d0 12 e0 4d d0 08  4d
8b68 : a2 aa a0 ac a9 05 d0 06  2f
8b70 : a2 c6 a0 ac a9 0a 20 f5  8a
8b78 : 8b ae 30 80 d0 08 a2 7f  4d
8b80 : a0 ac a9 07 d0 06 a2 87  98
8b88 : a0 ac a9 04 20 f5 8b 20  89
8b90 : b2 a0 bd 9a 8b 48 bd 99  7a
8b98 : 8b 48 60 b7 8b c2 8b c5  df
8ba0 : 8b c0 94 8e 85 cd 8b d0  19
8ba8 : 8b d3 8b db 8b de 8b e6  27
8bb0 : 8b e9 8b ec 8b 00 28 50  aa
8bb8 : a9 50 8d 31 80 a9 00 8d  83
8bc0 : 30 80 2c a9 01 2c a9 02  8d
8bc8 : 8d 2e 80 4c 23 8b a9 00  4b
8bd0 : 2c a9 01 2c a9 02 8d 2f  d6
8bd8 : 80 4c 23 8b a9 00 2c a9  57
8be0 : 6e 8d 30 80 4c 23 8b a9  90
8be8 : 50 2c a9 4d 2c a9 0f 8d  ca
8bf0 : 31 80 4c 23 8b 86 fd 84  c7
8bf8 : fe aa a0 00 b1 fd 20 d2  a5
8c00 : ff c8 ca d0 f7 60 20 40  b3
8c08 : 99 20 3b 9e a2 0f a0 13  9f
8c10 : 8e 38 80 8c 39 80 [illegible] 06  9a
8c18 : a0 01 20 a0 9d a9 [illegible] a0  05
8c20 : ac 20 5a 97 a2 0b 8e 78  14
8c28 : b4 bd f7 a1 9d 7a b4 bd  e9
8c30 : 02 a2 9d ba b4 bd 0d a2  f5
8c38 : 9d fa b4 ca d0 eb 8a a2  b5
8c40 : 69 a0 9b 20 8e 94 20 b2  58
8c48 : a0 bd 56 8c 48 bd 55 8c  cf
8c50 : 48 a9 00 8d 78 b4 60 7a  c2
8c58 : 8c fc 8d a2 8c b1 8f b9  22
8c60 : 8c 59 8e d5 8c ba 8f c9  67
8c68 : 8c b7 8f 6c 8c 20 32 9b  0b
8c70 : a9 01 8d 26 80 20 07 9b  1e
8c78 : 4c 8f 85 a2 15 a0 25 8e  4a
8c80 : 38 80 8c 39 80 a2 11 a0  e5
8c88 : 01 20 a0 9d a9 53 20 4b  c1
8c90 : 9b a9 a7 a0 af 20 5a 97  92
8c98 : 20 78 8d f0 03 20 61 9b  64
8ca0 : 4c 09 8c a9 52 20 4b 9b  54
8ca8 : a9 fe a0 af 20 99 8d f0  d5
8cb0 : ef 20 d9 8d 20 ea 8d 4c  ff
8cb8 : 9d 8c a9 43 20 4b 9b a9  8c
8cc0 : 66 a0 b0 20 99 8d f0 d8  22
8cc8 : d0 e7 a9 57 48 a9 4d 48  79
8cd0 : a9 88 a0 b0 d0 0a a9 41  82
8cd8 : 48 a9 52 48 a9 76 a0 b0  c5
8ce0 : 20 99 8d f0 bb a9 2c 99  3b
8ce8 : 04 b4 99 06 b4 a9 53 99  87
8cf0 : 05 b4 68 99 07 b4 98 18  45
8cf8 : 69 04 a2 04 a0 b4 20 bd  38
8d00 : ff a9 03 a2 08 a8 20 ba  a5
8d08 : ff 20 c0 ff 20 40 99 20  f2
8d10 : d9 8d ac 12 80 a9 2c 99  56
8d18 : f3 b3 99 f5 b3 a9 53 99  13
8d20 : f4 b3 68 99 f6 b3 98 18  db
8d28 : 69 04 a2 f3 a0 b3 20 bd  5e
8d30 : ff a9 04 a2 08 a8 20 ba  15
8d38 : ff 20 c0 ff 20 40 99 a9  35
8d40 : 00 85 90 20 cc ff a2 03  88
8d48 : 20 c6 ff 20 cf ff 8d 74  eb
8d50 : 8d 24 90 50 10 20 6b 8d  e8
8d58 : a9 03 20 c3 ff a9 04 20  a1
8d60 : c3 ff 4c 06 8c 20 6b 8d  89
8d68 : 4c 3f 8d 20 cc ff a2 04  1b
8d70 : 20 c9 ff a9 7b 4c d2 ff  0f
8d78 : a9 be a0 af 20 5a 97 a2  17
8d80 : 13 a0 01 18 20 f0 ff a2  f6
8d88 : f3 a0 b3 a9 25 20 ad 9e  35
8d90 : a9 0d ac 12 80 99 f3 b3  39
8d98 : 60 48 98 48 a2 16 a0 1e  e5
8da0 : 8e 38 80 8c 39 80 a2 10  3e
8da8 : a0 02 20 a0 9d 68 a8 68  f6
8db0 : 20 5a 97 a9 12 a0 b0 20  42
8db8 : 5a 97 a2 12 a0 0d 18 20  dc
8dc0 : f0 ff a9 3d 8d 03 b4 a2  cb
8dc8 : 04 a0 b4 a9 10 20 ad 9e  75
8dd0 : a9 0d ac 12 80 99 04 b4  bc
8dd8 : 60 a2 14 a0 0d 18 20 f0  96
8de0 : ff a2 f3 a0 b3 a9 10 4c  a3
8de8 : ad 9e a2 00 ac 12 80 bd  66
8df0 : 03 b4 99 f3 b3 c9 0d f0  d2
8df8 : df c8 e8 d0 f2 a2 15 a0  69
8e00 : 24 8e 38 80 8c 39 80 a2  63
8e08 : 11 a0 02 20 a0 9d a9 69  5e
8e10 : a0 ad 20 5a 97 a2 13 a0  f6
8e18 : 0d 18 20 f0 ff a9 4e 20  1e
8e20 : 4b 9b 20 3c 9f a9 10 a2  95
8e28 : f3 a0 b3 20 ad 9e 20 4a  41
8e30 : 9f ae 12 80 d0 03 4c 4c  1d
8e38 : 8c a2 13 a0 23 18 20 f0  44
8e40 : ff a9 02 a2 03 a0 b4 20  31
8e48 : ad 9e ae 12 80 f0 05 a9  29
8e50 : 2c 8d 02 b4 20 61 9b 4c  6e
8e58 : 06 8c a2 16 a0 1e 8e 38  b5
8e60 : 80 8c 39 80 a2 0f a0 02  ae
8e68 : 20 a0 9d 20 df 9d a9 ec  af
8e70 : a0 ad 20 5a 97 a9 7a a0  2c
8e78 : 06 ae 89 b3 30 02 a0 00  44
8e80 : 99 b3 06 a0 05 ae 8a b3  e0
8e88 : 30 02 a0 00 99 db 06 a2  b7
8e90 : 05 8e 78 b4 bd 27 a2 9d  6c
8e98 : 7a b4 bd 2b a2 9d ba b4  ac
8ea0 : bd 2f a2 9d fa b4 ca d0  73
8ea8 : eb 20 4b 9b a2 f3 a0 b3  9d
8eb0 : a9 10 20 ad 9e f0 13 0a  f1
8eb8 : aa bd c1 8e 48 bd c0 8e  16
8ec0 : 48 60 a6 8f a9 8f 9b 8f  78
8ec8 : 9e 8f ac 12 80 d0 03 4c  ce
8ed0 : 06 8c a9 2c 99 f3 b3 99  47
8ed8 : f5 b3 a9 52 99 f6 b3 a9  cf
8ee0 : 53 ae 89 b3 10 02 a9 55  c5
8ee8 : 99 f4 b3 98 18 69 04 a2  1d
8ef0 : f3 a0 b3 20 bd ff a9 05  b1
8ef8 : a2 08 a8 20 ba ff 20 c0  7a
8f00 : ff 20 40 99 a9 1a 80 30  71
8f08 : 7f 20 8e 9d a9 00 85 65  6a
8f10 : 85 64 85 63 a2 05 20 c6  f5
8f18 : ff a9 00 85 90 20 cf ff  e6
8f20 : e6 65 d0 06 e6 64 d0 02  87
8f28 : e6 63 24 90 70 33 ae 8a  4b
8f30 : b3 10 02 8d 20 d2 ff     d6
8f38 : 38 20 f0 ff e0 18 90 d9  81
8f40 : a2 00 86 d4 a9 89 a0 ae  e5
8f48 : 20 5a 97 20 cc ff 20 e4  96
8f50 : ff f0 f8 c9 03 f0 31 a9  0e
8f58 : 94 a0 ae 20 5a 97 4c 14  a8
8f60 : 8f a9 cd a0 ae 20 5a 97  d0
8f68 : 20 3e 82 20 8d af 38 20  c3
8f70 : 7e af 20 99 84 a2 00 86  6b
8f78 : c6 bd 00 01 f0 06 20 d2  a2
8f80 : ff e8 d0 f5 a5 c6 f0 fc  34
8f88 : a9 05 20 c3 ff 20 8e 9d  ab
8f90 : 20 86 82 20 6b 82 20 91  06
8f98 : 9a 4c 06 8c a9 00 2c a9  0a
8fa0 : 80 8d 8a b3 4c 6b 8e a9  ae
8fa8 : 00 2c a9 80 8d 89 b3 4c  c5
8fb0 : 6b 8e 20 03 95 4c 09 8c  c4
8fb8 : a9 ff 2c a9 00 8d 85 b3  8b
8fc0 : a2 16 a0 25 8e 38 80 8c  00
8fc8 : 39 80 a2 0e a0 01 20 a0  80
8fd0 : 9d a9 e9 a0 ae 20 5a 97  55
8fd8 : a2 05 8e 78 b4 bd 18 a2  8e
8fe0 : 9d 7a b4 bd 1d a2 9d ba  72
8fe8 : b4 bd 22 a2 9d fa b4 ca  72
8ff0 : d0 eb a9 24 20 4b 9b a2  b5
8ff8 : 14 a0 01 18 20 f0 ff a2  6f
9000 : f3 a0 b3 a9 25 20 ad 9e  ad
9008 : f0 15 0a aa bd 14 90 48  aa
9010 : bd 13 90 48 60 2d 90 20  96
9018 : 90 33 90 36 90 26 90 ac  03
9020 : 12 80 d0 29 4c 06 8c a9  46
9028 : 2a 8d f3 b3 d0 1f a9 50  d9
9030 : 2c a9 53 2c a9 55 2c a9  d4
9038 : 52 48 a9 24 20 4b 9b 68  39
9040 : 8d f5 b3 a9 2a 8d f3 b3  30
9048 : a9 3d 8d f4 b3 ad 85 b3  b8
9050 : d0 0a 20 3b 9e a2 06 a0  ed
9058 : 16 20 1d 91 a2 f0 a0 b3  93
9060 : a9 28 8e 37 99 8c 39 99  be
9068 : 8d 35 99 a9 00 8d 30 99  8c
9070 : a2 02 8e 78 b4 a9 07 a2  c0
9078 : 02 a0 06 20 9e 94 a9 08  95
9080 : a2 02 a0 07 20 a8 94 20  06
9088 : 07 99 20 ff 98 20 cf ff  2e
9090 : 20 cf ff ad 85 b3 f0 03  0d
9098 : 20 14 9c a9 00 85 90 20  cd
90a0 : 51 91 a5 90 d0 1b ad 85  dd
```

```
90a8 : b3 f0 0c a2 d5 a0 bc a9  d3
90b0 : 1e 20 78 9c 4c 9b 90 20  b4
90b8 : e6 90 c9 02 d0 1f 4c 9b  07
90c0 : 90 ae 85 b3 d0 17 a2 01  d1
90c8 : 8e 78 b4 a2 07 a0 02 18  c1
90d0 : 20 f0 ff a9 98 a0 af 20  2b
90d8 : 5a 97 20 b2 a0 20 79 94  76
90e0 : 20 40 99 4c 06 8c ae 87  9f
90e8 : b3 a0 08 18 20 f0 ff a0  bb
90f0 : 00 b9 d5 bc 20 d2 ff c8  04
90f8 : c0 1e 90 f5 ee 87 b3 ae  01
9100 : 87 b3 ec 88 b3 b0 03 a9  cd
9108 : 02 60 ae 86 b3 8e 87 b3  ec
9110 : 20 b2 a0 48 a2 06 a0 16  c3
9118 : 20 1d 91 68 60 8e 86 b3  34
9120 : 8e 87 b3 8c 88 b3 8a 48  d1
9128 : 98 aa a0 24 8e 38 80 8c  88
9130 : 39 80 68 aa a0 08 20 a0  25
9138 : 9d a2 09 a0 05 8e 38 80  23
9140 : 8c 39 80 a2 07 a0 02 20  9b
9148 : a0 9d a9 89 a0 af 4c 5a  c0
9150 : 97 20 ee 9a 20 cf ff 20  c7
9158 : cf ff 20 cf ff aa 20 cf  9e
9160 : ff 85 62 86 63 20 93 84  1a
9168 : a2 00 bd 00 01 f0 06 9d  65
9170 : d5 bc e8 d0 f5 e8 20 cf  be
9178 : ff f0 13 ac 85 b3 f0 08  13
9180 : e0 02 d0 04 c9 12 f0 03  0d
9188 : 9d d5 bc e8 d0 e8 a9 0d  71
9190 : 8d f2 bc 60 a2 91 a0 a8  5c
9198 : 20 27 95 20 14 9c a2 36  d2
91a0 : a0 b3 a9 0d 20 78 9c 20  9e
91a8 : 32 9b 20 16 9d a2 00 8a  77
91b0 : 9d 75 bd 9d 53 be 9d 2f  2b
91b8 : bf e8 e0 de 90 f2 a2 60  eb
91c0 : 8e 07 99 20 b2 94 a2 01  99
91c8 : 8e 2a 80 20 78 9d a0 00  86
91d0 : 8c 2b 80 ae 2a 80 e0 12  36
91d8 : f0 47 ac 2b 80 20 99 98  9d
91e0 : ae 2a 80 ac 2b 80 20 ed  6c
91e8 : 92 a0 00 b1 fd c9 02 f0  19
91f0 : 04 a9 ff 91 fd ac d6 ba  11
91f8 : ae d5 ba f0 08 c0 15 b0  9a
9200 : 04 e0 24 90 08 a9 01 a0  a3
9208 : 00 91 fd f0 09 20 ed 92  dd
9210 : a0 00 a9 02 91 fd ee 2b  76
9218 : 80 ac 2b 80 cc 16 80 90  6a
9220 : b2 ee 2a 80 ae 2a 80 e0  e4
9228 : 24 90 9d a2 03 ac 31 80  ab
9230 : c0 0f d0 02 a2 05 8e 8f  98
9238 : b3 8e 90 b3 a2 75 a0 bd  a1
9240 : 86 fd 84 fe a2 00 8e 8b  41
9248 : b3 e8 8e 2a 80 e0 12 f0  91
9250 : 5c 20 78 9d a0 00 8c 2b  21
9258 : 80 a0 00 b1 fd 10 3d ad  0f
9260 : 2a 80 20 f9 9a 8e 55 b3  ec
9268 : 8d 56 b3 ad 2b 80 20 f9  ee
9270 : 9a 8e 60 b3 8d 61 b3 a0  d4
9278 : 19 b9 4f b3 99 d5 bc 88  04
9280 : 10 f7 a2 d5 a0 bc a9 19  b8
9288 : 20 78 9c ee 8b b3 ce 90  9c
9290 : b3 d0 09 ae 8f b3 8e 90  b5
9298 : b3 20 5b 9d e6 fd d0 02  8b
92a0 : e6 fe ee 2b 80 ac 2b 80  41
92a8 : cc 16 80 90 ac ee 2a 80  9d
92b0 : ae 2a 80 e0 24 90 93 20  05
92b8 : 5b 9d ad 8b b3 d0 08 a2  e6
92c0 : 43 a0 b3 a9 0c d0 20 a2  82
92c8 : 00 20 8f 84 a9 20 8d 6a  f3
92d0 : b3 8d 6b b3 a2 00 bd 00  bc
92d8 : 01 f0 06 9d 69 b3 e8 d0  00
92e0 : f5 a2 69 a0 b3 a9 15 20  b2
92e8 : 78 9c 4c 8f 85 8e 8c b3  1a
92f0 : 8c 8d b3 a2 75 a0 bd 86  e4
92f8 : fd 84 fe ad 16 80 48 a2  78
9300 : 01 e0 12 f0 1b 20 78 9d  e4
9308 : a0 00 cc 8d b3 d0 05 ec  3d
9310 : 8c b3 f0 0f e6 fd d0 02  39
9318 : e6 fe c8 cc 16 80 d0 ea  c7
9320 : e8 d0 de 68 8d 16 80 60  81
9328 : a2 a0 a0 b0 20 27 95 20  2a
9330 : 14 9c a2 fb a0 b1 a9 13  1f
9338 : 20 78 9c 20 32 9b 20 16  6c
9340 : 9d a2 60 8e 07 99 20 b2  3b
9348 : 94 a2 02 ac 31 80 c0 0f  7c
9350 : d0 01 e8 8e 8f b3 8e 90  9f
9358 : b3 20 0b 94 ce 90 b3 d0  53
9360 : f8 20 5b 9d ae 8f b3 8e  46
9368 : 90 b3 20 68 9d ce 90 b3  e1
9370 : d0 f8 20 5b 9d ae 8f b3  25
9378 : 8e 90 b3 a2 00 8e 81 b3  71
9380 : e8 8e 2a 80 20 78 9d a0  c7
9388 : 00 8c 2b 80 ac 2b 80 ae  2d
9390 : 2a 80 a9 31 20 88 9a 20  7c
9398 : e0 98 ae 1a 80 10 32 ad  60
93a0 : 2a 80 20 f9 9a 8e f1 bc  b1
93a8 : 8d f2 bc ad 2b 80 20 f9  be
93b0 : 9a 8e f4 bc 8d f5 bc a2  27
93b8 : d5 a0 bc a9 27 20 78 9c  d0
93c0 : ee 81 b3 ce 90 b3 d0 09  31
93c8 : ae 8f b3 8e 90 b3 20 5b  da
93d0 : 9d ee 2b 80 ac 2b 80 cc  7f
93d8 : 16 80 90 b0 ee 2a 80 ae  08
93e0 : 2a 80 e0 24 90 9b 20 5b  24
93e8 : 9d ad 81 b3 d0 08 a2 0e  27
93f0 : a0 b2 a9 18 d0 0f 20 f9  51
93f8 : 9a 8e 26 b2 8d 27 b2 a2  db
9400 : 26 a0 b2 a9 14 20 78 9c  b5
9408 : 4c 8f 85 a2 41 a0 b4 a9  11
9410 : 27 4c 78 9c a2 14 a0 25  a7
9418 : 8e 38 80 8c 39 80 a2 11  b8
9420 : a0 01 20 a0 9d a9 6f a0  83
9428 : b2 20 5a 97 a2 3a a0 b2  58
9430 : 20 27 95 ad 2c 80 09 30  4a
9438 : 8d cb b3 a2 ca a0 b3 a9  c0
9440 : 13 20 6f 94 20 07 99 20  b3
9448 : 69 94 20 32 9b 20 b2 94  f8
9450 : a2 01 a0 00 20 b1 98 ae  ea
9458 : 2a 80 a0 00 20 96 98 a2  49
9460 : 23 ec 2a 80 d0 ec 4c 8f  59
9468 : 85 a2 c9 a0 b3 a9 01 8e  6e
9470 : 21 99 8c 23 99 8d 1f 99  9b
9478 : 60 a9 01 a2 c8 a0 b3 8d  bd
9480 : 35 99 8e 37 99 8c 39 99  22
9488 : a9 02 8d 30 99 60 8e 00  73
9490 : d0 8e 02 d0 8c 01 d0 8c  6f
9498 : 03 d0 8d 10 d0 60 8d 7b  a6
94a0 : b4 8e bb b4 8c fb b4 60  5d
94a8 : 8d 7c b4 8e bc b4 8c fc  10
94b0 : b4 60 a2 d5 a0 ba d0 04  23
94b8 : a2 d5 a0 b9 86 fb 84 fc  f8
94c0 : 60 a2 dd a0 b3 a9 13 20  12
94c8 : 6f 94 20 07 99 20 69 94  d4
94d0 : 20 0e 95 a2 2c a0 80 86  88
94d8 : fd 84 fe a2 36 a0 80 a9  e9
94e0 : fd 20 d8 ff a9 01 20 d5  f2
94e8 : ff 20 40 99 a2 12 ac 26  f4
94f0 : 80 b9 91 b3 a8 20 96 98  3f
94f8 : a9 20 8d 3e 05 4c 8f 85  d8
9500 : 20 07 99 20 fa 98 a9 56  d6
9508 : 20 d2 ff 4c 40 99 ad 2c  fb
9510 : 80 09 30 8d de b3 ae 2d  73
9518 : 80 a0 00 20 ba ff a9 12  63
9520 : a2 de a0 b3 4c bd ff 8e  a0
9528 : 3f 95 8c 41 95 a2 0e a0  65
9530 : 1e 8e 38 80 8c 39 80 a2  8d
9538 : 09 a0 0a 20 a0 9d a9 7b  ad
9540 : a0 7b 20 5a 97 a9 3a a0  e2
9548 : b1 20 5a 97 a9 0c a2 0e  34
9550 : a0 12 20 9e 94 a2 15 a0  c9
9558 : 1b 20 a8 94 a9 02 8d 78  12
9560 : b4 20 b2 a0 e0 02 d0 01  48
9568 : 60 68 68 4c 8f 85 48 a9  3a
9570 : 7f 10 08 48 a9 ff 30 03  64
9578 : 48 a9 00 8d 24 80 68 8e  4c
9580 : a7 95 8c a9 95 8d 09 96  61
9588 : 20 29 97 10 05 a9 02 8d  e5
9590 : 1c 80 a2 12 a0 24 8e 38  ad
9598 : 80 8c 39 80 a2 08 a0 03  b0
95a0 : 20 a0 9d 20 df 9d a9 7b  04
95a8 : a0 7b 20 5a 97 a9 58 a0  c3
95b0 : b1 20 5a 97 a2 7a ad 1b  e6
95b8 : 80 c9 08 f0 03 8e c5 05  03
95c0 : ad 1c 80 c9 08 f0 03 8e  06
95c8 : ca 05 ad 1d 80 c9 08 f0  7c
95d0 : 03 8e cf 05 ad 1e 80 c9  10
95d8 : 08 f0 03 8e d4 05 ad 1f  55
95e0 : 80 c9 08 f0 03 8e d9 05  7b
95e8 : ad 20 80 f0 03 8e 46 06  ad
95f0 : ad 21 80 d0 03 8e 40 06  1a
95f8 : ad 22 80 d0 03 8e 18 06  02
9600 : ad 23 80 f0 03 8e 1f 06  aa
9608 : a2 7b 8e 78 b4 bd 33 a2  66
9610 : 9d 7a b4 bd 3d a2 9d ba  a4
9618 : b4 bd 47 a2 9d fa b4 ca  eb
9620 : d0 eb a2 10 a0 0e 18 20  ac
9628 : f0 ff 20 4b 9b a2 f3 a0  69
9630 : b3 a9 10 20 ad 9e f0 1f  92
9638 : 0a aa bd 42 96 48 bd 41  74
9640 : 96 48 60 5c 96 99 96 87  3e
9648 : 96 ab 96 bd 96 f4 96 e2  42
9650 : 96 cf 96 08 97 17 97 ac  5e
9658 : 12 80 d0 01 60 ad 24 80  04
9660 : 10 0d 20 3f 97 f0 08 8c  21
9668 : 25 80 a9 07 8d 1f 80 d0  8e
9670 : 10 ad 21 80 49 40 8d 21  be
9678 : 80 ad 20 80 49 40 8d 20  f4
9680 : 80 a2 f3 a0 b3 4c dc 99  a6
9688 : 20 29 97 ad 1c 80 49 0a  d7
9690 : 8d 1c 80 20 39 97 d0 48  74
9698 : f0 f1 20 29 97 ad 1b 80  02
96a0 : 49 09 8d 1b 80 20 39 97  52
96a8 : d0 36 f0 f1 20 29 97 ad  13
96b0 : 1d 80 49 0b 8d 1d 80 20  c5
96b8 : 39 97 d0 24 f0 f1 20 29  e7
96c0 : 97 ad 1e 80 49 0c 8d 1e  2d
96c8 : 80 20 39 97 d0 12 f0 f1  df
96d0 : 20 29 97 ad 1f 80 49 0f  5a
96d8 : 8d 1f 80 20 39 97 f0 f3  15
96e0 : 4c a3 95 20 4c 97 ad 20  e0
96e8 : 80 49 40 8d 20 80 d0 f0  fa
96f0 : ad 21 80 f0 eb 20 4c 97  8c
96f8 : ad 21 80 49 40 8d 21 80  75
9700 : f0 de ad 20 80 d0 d9 f0  a7
9708 : e0 ad 22 80 49 80 8d 22  6b
9710 : 80 f0 cd ad 23 80 d0 c8  3c
9718 : ad 23 80 49 80 8d 23 80  a2
9720 : d0 be ad 22 80 f0 b9 d0  17
9728 : e3 ad 24 80 10 1d a0 04  6f
9730 : a9 08 99 1b 80 88 10 fa  2a
9738 : 60 29 f8 c9 08 d0 0c a0  1d
9740 : 05 b9 1a 80 c9 08 d0 03  df
9748 : 88 d0 f6 60 ad 24 80 d0  a2
9750 : fa 8d 20 80 a9 40 8d 21  3e
9758 : 80 60 85 fd 84 fe a0 00  ec
9760 : b1 fd f0 0a 20 d2 ff c8  b7
9768 : d0 f6 e6 fe d0 f2 60 ad  ce
9770 : 32 80 f0 fa 8e 17 80 8c  3a
9778 : 18 80 a5 fb 48 a5 fc 48  ef
9780 : 20 b2 94 a2 12 a0 00 20  d9
9788 : 99 98 a0 00 b9 d5 bb d9  82
9790 : d5 ba d0 06 c8 d0 f5 4c  3b
9798 : 01 98 a5 fd 48 a5 fe 48  4d
97a0 : 20 0e 98 a2 0d a0 1d 8e  a9
97a8 : 38 80 8c 39 80 a2 08 a0  e9
97b0 : 09 20 a0 9d a9 6f a0 a5  89
97b8 : 20 5a 97 a2 01 86 d4 bd  53
97c0 : 64 bb 20 d2 ff e8 e0 11  51
97c8 : d0 01 e8 e0 18 90 f0 a2  7e
97d0 : 00 86 d4 a9 0c aa a0 10  36
97d8 : 20 9e 94 a2 16 a0 1c 20  d8
97e0 : a8 94 a9 02 8d 78 b4 20  2d
97e8 : b2 a0 c9 04 f0 0d c9 02  80
97f0 : d0 f5 4c 7a 98 68 85 fe  fe
97f8 : 68 85 fd 20 50 82 20 44  c8
9800 : 98 68 85 fc 68 85 fb ac  c9
9808 : 18 80 ae 17 80 60 a2 17  b3
9810 : bd 20 05 9d 12 b9 bd 48  49
9818 : 05 9d 2a b9 bd 70 05 9d  5c
9820 : 42 b9 bd 98 05 9d 5a b9  db
9828 : bd c0 05 9d 72 b9 bd e8  f8
9830 : 05 9d 8a b9 bd 10 06 9d  8d
9838 : a2 b9 bd 38 06 9d ba b9  d9
9840 : ca 10 cd 60 a2 17 bd 12  90
9848 : b9 9d 20 05 bd 2a b9 9d  c8
9850 : 48 05 bd 42 b9 9d 70 05  27
9858 : bd 5a b9 9d 98 05 bd 72  f2
9860 : b9 9d c0 05 bd 8a b9 9d  0b
9868 : e8 05 bd a2 b9 9d 10 06  6b
9870 : bd ba b9 9d 38 06 ca 10  ab
9878 : cd 60 20 32 9b 20 b8 94  8a
9880 : ae 17 80 e0 12 d0 0c a0  0f
9888 : 00 8c 26 80 c8 8c 18 80  ba
9890 : 20 99 98 4c f5 97 20 6f  a8
9898 : 97 a9 31 20 88 9a 20 e0  f4
98a0 : 98 20 ff 98 a0 00 20 cf  85
98a8 : ff 91 fb c8 d0 f8 4c 40  0e
98b0 : 99 20 6f 97 a9 32 20 88  e6
98b8 : 9a 20 07 99 20 fa 98 a0  d5
98c0 : 08 a2 00 bd c0 b3 20 d2  a1
98c8 : ff e8 88 d0 f6 20 cc ff  1b
98d0 : 20 f7 98 a0 00 b1 fb 20  e4
98d8 : d2 ff c8 d0 f8 20 cc ff  ba
98e0 : 20 07 99 20 fa 98 a0 0d  ff
98e8 : a2 00 bd b3 b3 20 d2 ff  f8
98f0 : e8 88 d0 f6 4c 40 99 a2  a2
98f8 : 02 2c a2 0f 4c c9 ff a2  f3
9900 : 02 2c a2 0f 4c c6 ff ea  74
9908 : a9 0f 20 c3 ff a9 02 20  4f
9910 : c3 ff 20 cc ff a9 0f ae  5b
9918 : 2d 80 a8 20 ba ff a9 01  08
9920 : a2 c9 a0 b3 20 bd ff 20  76
9928 : c0 ff a9 02 ae 2d 80 a0  2a
9930 : 02 20 ba ff a9 01 a2 c8  b0
9938 : a0 b3 20 bd ff 4c c0 ff  d7
9940 : 20 cc ff ad 78 b4 48 a9  1d
9948 : 00 8d 78 b4 a9 20 a2 17  18
9950 : 9d d0 07 ca d0 fa 20 02  da
9958 : 99 20 cf ff 48 c9 30 f0  6b
9960 : 03 a2 ff 2c a2 00 8e 1a  d2
9968 : 80 a2 18 a0 11 18 20 f0  88
9970 : ff 20 ee 9a 68 8d d5 bc  52
9978 : 20 d2 ff 20 cf ff 8d d6  e6
9980 : bc 20 d2 ff 20 cf ff 8d  9c
9988 : d7 bc 20 d2 ff a0 00 20  65
9990 : cf ff c9 2c f0 0d c0 13  f7
```

Listing. »XDOS« (Fortsetzung)

```
9998 : b0 03 20 d2 ff 99 d8 bc  d6
99a0 : c8 d0 ec 20 cf ff c9 0d  4e
99a8 : d0 f9 ae 1a 80 10 26 20  c5
99b0 : 0e 98 a2 0e a0 1c 8e 38  0a
99b8 : 80 8c 39 80 a2 08 a0 0a  de
99c0 : 20 a0 9d a9 cc a0 a5 20  75
99c8 : 5a 97 20 53 9f 30 fb 20  0c
99d0 : bb a0 20 44 98 68 8d 78  60
99d8 : b4 4c cc ff 86 fd 84 fe  4e
99e0 : a0 00 b1 fd c9 a0 f0 07  20
99e8 : c9 2a f0 0e c8 d0 f3 c0  29
99f0 : 00 d0 05 a9 2a 91 fd 2c  4e
99f8 : a0 0f 8c 28 80 a2 00 8e  82
9a00 : 27 80 e8 8e 26 80 20 07  68
9a08 : 9b ae 27 80 bd a3 b3 aa  f1
9a10 : a0 03 bd d5 b9 29 07 d9  11
9a18 : 1b 80 f0 0a 88 10 f3 49  5c
9a20 : 07 cd 1f 80 d0 3b bd d5  6f
9a28 : b9 29 40 cd 20 80 f0 05  13
9a30 : cd 21 80 d0 2c bd d5 b9  43
9a38 : 29 80 cd 23 80 f0 05 cd  b8
9a40 : 22 80 d0 1d e8 e8 bd d6  f5
9a48 : b9 29 7f f0 38 a0 ff c8  ae
9a50 : e8 cc 28 80 b0 27 b1 fd  bf
9a58 : c9 3f f0 f3 dd d5 b9 f0  d1
9a60 : ee ee 27 80 ae 27 80 e0  87
9a68 : 08 90 9e ad d6 b9 30 15  3c
9a70 : a2 00 8e 27 80 20 20 9b  5c
9a78 : b0 0b 4c 09 9a ae 27 80  9f
9a80 : bd a3 b3 aa 60 a2 00 60  2d
9a88 : 8e 2a 80 8c 2b 80 8d b4  33
9a90 : b3 ad 2c 80 09 30 8d b8  ef
9a98 : b3 8d cb 07 ad 2a 80 20  54
9aa0 : f9 9a 8e ba b3 8d bb b3  df
9aa8 : 8e c1 07 8d c2 07 ad 2b  fc
9ab0 : 80 20 f9 9a 8e bd b3 8d  d3
9ab8 : be b3 8e c5 07 8d c6 07  b2
9ac0 : ad 2d 80 20 f9 9a 8e c8  68
9ac8 : 07 8d c9 07 ad 33 80 09  72
9ad0 : 30 8d cd 07 a9 19 ae 35  a4
9ad8 : 80 f0 02 a9 0e 8d cf 07  21
9ae0 : ad 32 80 30 03 a9 20 2c  23
9ae8 : a9 1f 8d ee 04 60 a9 20  8c
9af0 : a0 4f 99 d5 bc 88 10 fa  9f
9af8 : 60 d8 a2 30 38 e9 0a 90  8f
9b00 : 03 e8 b0 f9 69 3a 60 ac  26
9b08 : 26 80 f0 12 88 ce 26 80  85
9b10 : b9 91 b3 20 b8 94 a8 a2  9b
9b18 : 12 20 96 98 18 24 38 60  37
9b20 : ac 26 80 c0 11 b0 f8 ad  ed
9b28 : d6 b9 30 f2 c8 8c 26 80  d0
9b30 : d0 de 20 38 9b 4c 99 98  32
9b38 : a2 d5 a0 bb 86 fb 84 fc  b9
9b40 : a2 12 a0 00 60 20 38 9b  33
9b48 : 4c b4 98 8d f0 b3 ad 2c  82
9b50 : 80 09 30 8d f1 b3 a9 a0  b7
9b58 : a2 24 9d f3 b3 ca 10 fa  ba
9b60 : 60 20 fa 98 a0 00 b9 f0  75
9b68 : b3 20 d2 ff c9 0d f0 03  af
9b70 : c8 d0 f3 4c 40 99 a2 05  8c
9b78 : a0 12 18 20 f0 ff a2 90  e6
9b80 : 86 d4 bd d5 bb 20 d2 ff  a2
9b88 : e8 e0 a5 90 f5 a2 00 86  dd
9b90 : d4 ca ad f5 06 c9 20 f0  04
9b98 : 0c ad 29 80 4a 4a 4a 4a  8a
9ba0 : 4a 49 07 aa e8 8e 83 b3  1e
9ba8 : a2 08 8e 82 b3 ca a0 12  7a
9bb0 : 20 ee 9b a9 20 20 d2 ff  b2
9bb8 : ad 83 b3 cd 82 b3 f0 03  5d
9bc0 : a9 00 2c a9 20 8d 84 b3  91
9bc8 : ad 39 30 29 07 0a 0a a8  7d
9bd0 : 8a 48 a2 03 b9 2d b4 0d  79
9bd8 : 84 b3 20 d2 ff c8 ca 10  2a
9be0 : f3 68 aa e8 ce 82 b3 d0  40
9be8 : c5 a9 00 85 d4 60 18 20  24
9bf0 : f0 ff bd 61 b4 8d 09 9c  90
9bf8 : 8d c9 9b bd 69 b4 8d 0a  8f
9c00 : 9c 8d ca 9b a0 03 84 d4  67
9c08 : b9 39 30 20 d2 ff c8 c0  40
9c10 : 13 90 f5 60 ad 2f 80 f0  2d
9c18 : 1c c9 02 f0 0f a9 ff 8d  11
9c20 : 03 dd ad 00 dd 09 04 8d  cf
9c28 : 00 dd d0 1f a9 9c 48 a9  23
9c30 : 4a 48 6c 09 80 a9 04 20  80
9c38 : c3 ff a9 04 aa ac 34 80  c8
9c40 : 20 ba ff a9 00 20 bd ff  ea
9c48 : 20 c0 ff ad 2e 80 c9 01  8e
9c50 : f0 05 c9 02 f0 15 60 a9  02
9c58 : 40 20 6e 9c ad 30 80 20  76
9c60 : 6e 9c ad 31 80 20 6e 9c  aa
9c68 : 4c 5b 9d 6c 0b 80 48 a9  80
9c70 : 1b 20 bd 9c 68 4c bd 9c  b7
9c78 : 8d 14 80 a5 fd 48 a5 fe  9b
9c80 : 48 86 fd 84 fe a0 00 8c  29
9c88 : 15 80 b1 fd c9 0d d0 06  5e
9c90 : 20 5b 9d 4c 99 9c 20 a9  a1
```

```
9c98 : 9c ac 15 80 c8 cc 14 80  24
9ca0 : d0 e5 68 85 fe 68 85 fd  73
9ca8 : 60 c9 20 90 0a c9 80 90  19
9cb0 : 0c c9 a0 f0 06 b0 06 18  15
9cb8 : 69 40 2c a9 20 ae 2e 80  b3
9cc0 : f0 24 ca d0 18 c9 41 90  85
9cc8 : 1d c9 5a 90 0b c9 c1 90  9a
9cd0 : 15 c9 db b0 11 38 e9 a0  92
9cd8 : 18 69 20 d0 09 a9 9c 48  a8
9ce0 : a9 e5 48 6c 0d 80 ae 2f  09
9ce8 : 80 d0 0d 48 a2 04 20 c9  7b
9cf0 : ff 68 20 d2 ff 4c cc ff  1b
9cf8 : ca d0 18 8d 01 dd a9 10  a8
9d00 : 2c 0d dd f0 fb ad 00 dd  31
9d08 : 29 fb 8d 00 dd 09 04 8d  e4
9d10 : 00 dd 60 6c 0f 80 20 5b  d1
9d18 : 9d a2 0f bd 65 bc 29 7f  62
9d20 : 9d e9 b2 ca 10 f5 a2 04  fb
9d28 : bd 77 bc 29 7f 9d ff b2  3f
9d30 : ca 10 f5 a2 de a0 b2 a9  e5
9d38 : 35 20 78 9c ad d7 bb c9  4b
9d40 : 43 f0 0c a2 13 a0 b3 a9  ab
9d48 : 16 20 78 9c 4c 58 9d a2  63
9d50 : 29 a0 b3 a9 0d 20 78 9c  d8
9d58 : 20 5b 9d a9 0d 20 bd 9c  c4
9d60 : ad 35 80 f0 12 4c bd 9c  9a
9d68 : a0 27 8c 8e b3 a9 2d 20  0e
9d70 : a9 9c ce 8e b3 d0 f6 60  4b
9d78 : a0 15 e0 12 90 0c a0 13  2f
9d80 : e0 19 90 06 88 e0 1f 90  ff
9d88 : 01 88 8c 16 80 60 20 5b  f5
9d90 : ff a9 0f 8d 20 d0 8d 21  da
9d98 : d0 a9 52 a0 a2 4c 5a 97  0b
9da0 : 8e 36 80 8c 37 80 ca 88  af
9da8 : 8e 3d 80 8c 3e 80 ac 39  94
9db0 : 80 c8 8c 3f 80 a9 70 a2  fc
9db8 : 40 a0 6e 20 e9 9d ee 3d  aa
9dc0 : 80 ae 3d 80 ec 38 80 b0  eb
9dc8 : 08 a9 5d a2 20 a0 f5 d0  d1
9dd0 : ea a9 6d a2 f8 a0 fe 20  0f
9dd8 : e9 9d a9 00 8d 43 80 ae  4d
9de0 : 36 80 ac 37 80 18 4c f0  44
9de8 : ff 8d 3a 80 8e 3b 80 8c  2a
9df0 : 3c 80 ae 3d 80 8e 40 80  3e
9df8 : 20 c3 a0 ac 3e 80 ad 3a  cb
9e00 : 80 91 fd ad 3f 80 38 ed  33
9e08 : 3e 80 aa c8 ad 3b [illegible] 91  24
9e10 : fd c8 ca d0 fa ad 3c 80  4d
9e18 : 91 fd 60 a2 00 bd c8 04  2d
9e20 : 29 7f 9d 7a b5 bd b8 05  f6
9e28 : 29 7f 9d 6a b6 bd a8 06  cd
9e30 : 29 7f 9d 5a b7 e8 e0 f0  f4
9e38 : d0 e3 60 a2 00 bd 7a b5  a9
9e40 : 9d c8 04 bd 6a b6 9d b8  3e
9e48 : 05 bd 5a b7 9d a8 06 e8  c2
9e50 : e0 f0 d0 e9 60 a9 20 8d  09
9e58 : ea 06 8d f5 06 8d 45 07  57
9e60 : a2 01 9d bf 06 9d 10 07  7e
9e68 : 9d 13 07 9d 19 07 9d 1c  7d
9e70 : 07 9d 5f 07 ca 10 eb a2  21
9e78 : 02 9d c1 06 9d cd 06 9d  16
9e80 : 61 07 9d 6d 07 9d 8b 07  13
9e88 : 9d 95 07 9d 51 a7 9d 57  dd
9e90 : a7 ca 10 e5 a2 ca 9d 74  3d
9e98 : bd ca d0 fa a9 28 8d ab  b7
9ea0 : bd 8d 18 be a9 29 8d b1  7f
9ea8 : bd 8d 1e be 60 8d 11 80  43
9eb0 : 8e 13 80 8c 14 80 a0 00  41
9eb8 : 8c 12 80 8c 43 80 a9 a4  27
9ec0 : 20 14 9f 78 20 53 9f 58  ad
9ec8 : 10 54 20 e4 ff ac 12 80  56
9ed0 : c9 0d f0 48 c9 14 d0 1a  1a
9ed8 : c0 00 f0 e2 ce 12 80 48  41
9ee0 : 20 12 9f 68 20 d2 ff a9  ea
9ee8 : 94 20 d2 ff 20 12 9f 4c  eb
9ef0 : be 9e cc 11 80 b0 c7 c9  93
9ef8 : 22 f0 c3 8d 05 9f 29 7f  26
9f00 : c9 20 90 ba a9 7b ee 12  ab
9f08 : 80 99 25 bd 20 d2 ff 4c  87
9f10 : be 9e a9 20 20 d2 ff a9  78
9f18 : 9d 4c d2 ff a9 00 48 20  8c
9f20 : 12 9f ae 13 80 ac 14 80  cf
9f28 : 86 fd 84 fe ac 12 80 f0  ed
9f30 : 09 88 b9 25 bd 91 fd 88  02
9f38 : 10 f8 68 60 48 ad 78 b4  28
9f40 : 8d 79 b4 a9 00 8d 78 b4  a4
9f48 : 68 60 48 ad 79 b4 8d 78  0c
9f50 : b4 68 60 a9 e0 8d 02 dc  c2
9f58 : a9 03 8d 15 d0 a2 04 20  fb
9f60 : 7f a0 ad 00 dc 8d 44 80  e7
9f68 : 29 01 d0 0d ae 01 d0 e0  e0
9f70 : 32 90 e0 ce 01 d0 ce 03  d4
9f78 : d0 ad 44 80 29 02 d0 0d  40
9f80 : ae 01 d0 e0 f8 b0 cc ee  25
9f88 : 01 d0 ee 03 d0 ad 44 80  9a
9f90 : 29 04 d0 20 ae 10 d0 d0  44
```

```
9f98 : 0f ae 00 d0 e0 18 90 b3  91
9fa0 : ce 00 d0 ce 02 d0 d0 0c  7e
9fa8 : ae 00 d0 d0 f3 a2 00 8e  16
9fb0 : 10 d0 f0 ec ad 44 80 29  53
9fb8 : 08 d0 1e ae 10 d0 d0 0f  6f
9fc0 : ee 00 d0 ee 02 d0 d0 11  cc
9fc8 : a2 03 8e 10 d0 d0 0a ae  ab
9fd0 : 00 d0 e0 56 90 ea 4c 53  73
9fd8 : 9f ae 78 b4 d0 05 8e 42  77
9fe0 : 80 f0 6b ad 01 d0 38 e9  b4
9fe8 : 02 4a 4a 4a 38 e9 06 8d  f1
9ff0 : 40 80 ad 40 80 dd 7a b4  2e
9ff8 : f0 05 ca d0 f5 f0 df 8e  bb
a000 : 42 80 ad 00 d0 4a 4a 4a  0b
a008 : 38 e9 02 8d 41 80 ad 10  56
a010 : d0 29 01 f0 09 a9 20 18  62
a018 : 6d 41 80 8d 41 80 ad 41  49
a020 : 80 dd ba b4 90 07 dd fa  83
a028 : b4 90 07 f0 05 ca d0 c2  74
a030 : f0 ac 20 c3 a0 ae 42 80  80
a038 : ec 43 80 f0 19 bd ba b4  d8
a040 : a8 b1 fd 09 80 91 fd c8  7f
a048 : 98 dd fa b4 d0 f2 ad 42  04
a050 : 80 cd 43 80 d0 32 ad 42  72
a058 : 80 8d 43 80 ad 44 80 29  d1
a060 : 10 d0 0f a2 20 20 7f a0  33
a068 : ad 00 dc 29 10 f0 f9 a9  35
a070 : 00 2c a9 80 0d 42 80 08  f6
a078 : a2 ff 8e 02 dc 28 60 a0  d0
a080 : ff 88 d0 fd ca d0 f8 60  8f
a088 : ae 43 80 f0 c9 bd 7a b4  f4
a090 : 8d 40 80 bd ba b4 8d 41  1f
a098 : 80 20 c3 a0 ae 43 80 ac  8e
a0a0 : 41 80 b1 fd 29 7f 91 fd  1e
a0a8 : c8 98 dd fa b4 f0 a7 a8  56
a0b0 : d0 f0 20 53 9f f0 fb 30  3d
a0b8 : f9 0a aa 48 a9 00 8d 15  65
a0c0 : d0 68 60 ad 40 80 0a aa  18
a0c8 : bd d3 a0 85 fd bd d4 a0  aa
a0d0 : 85 fe 60 00 04 28 04 50  1f
a0d8 : 04 78 04 a0 04 c8 04 f0  a6
a0e0 : 04 18 05 40 05 68 05 90  03
a0e8 : 05 b8 05 e0 05 08 06 30  b0
a0f0 : 06 58 06 80 06 a8 06 d0  13
a0f8 : 06 f8 06 20 07 48 07 70  b0
a100 : 07 98 07 c0 07 05 06 07  ec
a108 : 09 0a 0c 0d 0e 0f 11 12  7d
a110 : 13 15 16 10 05 05 07 08  da
a118 : 09 0a 0b 0c 0d 0e 10 07  fa
a120 : 08 09 0a 0b 0c 0d 0e 10  12
a128 : 12 12 18 18 18 18 15 04  eb
a130 : 07 03 04 01 04 01 01 01  28
a138 : 04 02 01 01 01 20 12 24  40
a140 : 24 24 24 24 24 24 24 24  40
a148 : 1b 12 12 12 12 12 12 12  51
a150 : 12 12 1a 25 08 0b 0d 0f  c2
a158 : 0a 0e 0a 0f 0e 11 0d 10  8c
a160 : 11 11 0d 0e 10 07 10 27  c7
a168 : 22 26 27 27 27 27 27 27  e3
a170 : 27 27 1f 22 22 22 22 22  37
a178 : 22 22 22 1a 1b 26 0a 0c  9a
a180 : 0e 10 0f 0c 0c 0c 0c 0e  49
a188 : 0e 0e 0e 10 10 18 1b 1e  8d
a190 : 22 18 1b 1e 22 05 0b 1b  f6
a198 : 1e 22 26 1b 1e 22 26 09  52
a1a0 : 11 09 0a 0b 0b 0c 0d 0e  7b
a1a8 : 0f 11 11 13 13 14 16 16  3d
a1b0 : 16 22 22 16 16 16 16 1d  c7
a1b8 : 16 1c 16 21 1b 1a 26 26  ee
a1c0 : 1a 20 1a 1c 22 1a 21 20  ac
a1c8 : 07 08 0a 0a 0b 0b 0b 0b  e3
a1d0 : 0c 0c 0e 0e 0f 0f 10 10  f2
a1d8 : 1f 1f 17 1a 17 1a 1d 21  89
a1e0 : 17 1b 16 1b 17 1b 17 1d  4f
a1e8 : 23 23 1a 1d 1a 1d 21 25  20
a1f0 : 1a 1f 1b 1f 1a 1e 18 1e  74
a1f8 : 08 08 09 09 0a 0a 0b 0b  9b
a200 : 0c 0c 0e 01 0c 01 0c 01  b1
a208 : 0c 01 0c 01 0c 08 0a 14  09
a210 : 0a 14 0a 14 0a 14 0a 15  bd
a218 : 0e 10 10 10 10 12 11 16  37
a220 : 1b 20 15 16 1b 20 25 23  e1
a228 : 11 11 12 12 0b 12 0b 11  18
a230 : 10 16 10 19 10 0b 0b 0b  0e
a238 : 0b 0e 0e 0b 0d 0d 20 0d  03
a240 : 12 17 1c 10 17 21 10 18  d2
a248 : 24 12 17 1c 21 15 20 26  46
a250 : 15 1e 90 93 09 0e 08 00  2c
a258 : 13 20 20 20 20 20 20 20  4b
a260 : 20 2a 2a 2a 20 c3 2d 36  a6
a268 : 34 20 d8 c4 cf d3 a0 56  46
a270 : 31 2e 32 37 20 2a 2a 2a  7c
a278 : 0d 20 20 20 20 20 20 20  65
a280 : c3 4f 50 59 52 49 47 48  47
a288 : 54 20 28 c3 29 20 31 39  3a
a290 : 38 37 2c 20 36 34 27 45  9f
```

```
a298 : 52 0d 20 20 20 20 20 20  41
a2a0 : 20 cd 41 52 4b 54 20 26  66
a2a8 : 20 d4 45 43 48 4e 49 4b  9f
a2b0 : 20 d6 45 52 4c 41 47 20  03
a2b8 : c1 c7 0d 20 20 20 20 20  68
a2c0 : 20 d7 52 49 54 54 45 4e  23
a2c8 : 20 42 59 20 d0 45 54 45  77
a2d0 : 52 20 d7 49 4d 4d 45 52  4a
a2d8 : 20 4a 55 4e 2e 0d c0 c0  0c
a2e0 : c0 c0 c0 c0 c0 c0 c0 c0  df
a2e8 : c0 c0 c0 c0 c0 c0 c0 b2  cb
a2f0 : 0d 20 20 20 20 d8 c4 49  88
a2f8 : 52 45 43 54 4f 52 59 20  75
a300 : 20 20 dd 0d 20 20 20 20  0d
a308 : 20 20 20 c2 c1 cd 20 20  e4
a310 : 20 20 20 20 20 ab 0d 20  20
a318 : 20 20 c8 49 44 44 45 4e  bc
a320 : 20 c6 49 4c 45 53 20 20  2f
a328 : dd 0d 20 20 20 20 20 20  5c
a330 : 20 20 20 20 20 20 20 20  30
a338 : 20 20 20 dd 0d 20 20 20  bf
a340 : 20 c3 48 45 43 4b 20 c4  95
a348 : 49 53 4b 20 20 20 dd 0d  a6
a350 : 20 d0 45 52 46 4f 52 4d  37
a358 : 41 4e 43 45 20 d4 45 53  9e
a360 : 54 dd 0d 20 20 20 20 20  ae
a368 : 20 20 20 20 20 20 20 20  68
a370 : 20 20 20 20 dd 0d 20 20  b3
a378 : 20 20 d2 45 53 43 52 41  21
a380 : 54 43 48 20 20 20 20 dd  cb
a388 : 0d 20 d3 43 52 41 54 43  0a
a390 : 48 20 d0 52 4f 54 45 43  9a
a398 : 54 20 dd 0d 20 c4 49 53  09
a3a0 : 4b 20 d7 52 49 54 45 20  c8
a3a8 : d0 52 54 43 54 dd 0d 20  c7
a3b0 : c4 49 53 4b 20 c1 44 44  01
a3b8 : 52 20 c3 48 41 4e 47 45  42
a3c0 : ab 0d 20 20 20 20 20 20  c2
a3c8 : 20 20 20 20 20 20 20 20  c8
a3d0 : 20 20 20 dd d0 52 45 56  25
a3d8 : 49 4f 55 53 20 ce 45 58  c7
a3e0 : 54 20 ce 45 57 2e 44 53  3f
a3e8 : 4b 0d 20 20 20 20 c4 c9  70
a3f0 : d3 cb 20 cd c5 ce d5 20  d5
a3f8 : 20 20 20 dd c1 44 52 3a  e8
a400 : 20 20 20 20 20 20 20 c2  46
a408 : 4c 4f 43 4b 53 3a 0d 20  b2
a410 : 20 d0 52 49 4e 54 45 52  97
a418 : 20 cd 45 4e 55 20 20 20  51
a420 : dd c3 4c 4f 53 45 44 3a  c1
a428 : 20 20 20 d0 52 4f 54 45  f6
a430 : 43 54 3a 0d 20 d3 59 53  7a
a438 : 54 45 4d 20 d3 45 54 54  e8
a440 : 49 4e 47 53 20 dd 31 53  49
a448 : 54 20 d4 3a 20 20 28 20  0d
a450 : 20 29 20 d3 3a 20 20 28  fd
a458 : 20 20 29 0d 20 20 20 20  38
a460 : 20 20 20 20 20 20 20 20  60
a468 : 20 20 20 20 20 dd 20 d3  be
a470 : 49 44 45 20 d3 45 43 54  4e
a478 : 4f 52 20 c2 4c 4b 53 3a  32
a480 : 0d 20 20 d1 55 49 54 20  11
a488 : 20 20 20 c2 52 45 41 4b  04
a490 : 20 20 dd d2 45 43 53 3a  c2
a498 : 20 20 20 20 20 20 cc 45  95
a4a0 : 4e 47 54 48 3a 0d 20 d2  e2
a4a8 : 45 54 55 52 4e 20 54 4f  8d
a4b0 : 20 c2 c1 d3 c9 c3 20 dd  13
a4b8 : c4 53 4b 46 52 45 45 3a  9a
a4c0 : 20 20 20 20 20 d5 53 45  85
a4c8 : 44 3a 0d c0 c0 c0 c0 c0  1b
a4d0 : c0 c0 b2 c0 c0 c0 c0 b2  2f
a4d8 : c0 c0 c0 b2 b1 0d 54 1d  8e
a4e0 : 1d 20 53 1d 1d dd 1d 1d  f5
a4e8 : 20 1d dd 1d 20 1d dd 00  14
a4f0 : 00 00 00 40 00 00 30 00  b9
a4f8 : 00 3c 00 00 1f 00 00 1c  41
a500 : 00 00 0a 00 00 09 00 00  cb
a508 : 00 80 00 00 40 00 00 00  4d
a510 : 00 00 00 00 00 00 00 00  11
a518 : 00 00 00 00 00 00 00 00  19
a520 : 00 00 00 00 00 00 00 00  21
a528 : 00 00 00 00 00 00 00 00  29
a530 : e0 00 00 b8 00 00 ce 00  63
a538 : 00 43 80 00 60 80 00 23  4a
a540 : 80 00 35 80 00 16 c0 00  d1
a548 : 1f 60 00 01 a0 00 00 e0  83
a550 : 00 00 00 00 00 00 00 00  51
a558 : 00 00 00 00 00 00 00 00  59
a560 : 00 00 00 00 00 00 00 00  61
a568 : 00 00 00 00 00 00 00 20  a9
a570 : 20 21 20 c4 49 53 4b 20  5e
a578 : 45 58 43 48 41 4e 47 45  f1
a580 : 44 20 21 0d 1d 1d 1d 1d  28
a588 : 1d 1d 1d 1d 1d 1d 1d 1d  88
a590 : 1d a3 a3 a3 a3 b7 b7 b7  22
a598 : b7 b7 b7 a3 a3 a3 a3 0d  8d
a5a0 : 11 11 1d 1d 1d 1d 1d 1d  8e
a5a8 : 1d 1d 1d 1d 1d 1d c9 4e  bd
a5b0 : 49 54 20 20 20 20 20 20  f3
a5b8 : c9 47 4e 4f 52 45 91 91  5b
a5c0 : 91 0d 1d 1d 1d 1d 1d 1d  2c
a5c8 : 1d 1d 1d 00 20 20 20 21  81
a5d0 : 20 c4 49 53 4b 20 c5 52  81
a5d8 : 52 4f 52 20 21 11 9d 9d  b7
a5e0 : 9d 9d 9d 9d 9d 9d 9d 9d  df
a5e8 : 9d 9d a3 a3 a3 b7 b7 b7  f7
a5f0 : b7 a3 a3 a3 0d 1d 1d 1d  3f
a5f8 : 1d 1d 1d 1d 1d 1d 1d 1d  f8
a600 : d0 4c 45 41 53 45 20 43  d6
a608 : 48 45 43 4b 20 44 52 49  2d
a610 : 56 45 0d 1d 1d 1d 1d 1d  59
a618 : 1d 1d 1d 1d 1d 1d 41 4e  0b
a620 : 44 20 49 4e 53 45 52 54  e2
a628 : 20 41 20 44 49 53 4b 2e  32
a630 : 11 11 9d 9d 9d 9d 9d 9d  5d
a638 : 9d 9d 9d 9d 9d 9d 9d 9d  37
a640 : d0 52 45 53 53 20 c6 49  d9
a648 : 52 45 00 c5 58 54 45 4e  cf
a650 : 44 45 44 20 c4 49 52 45  b6
a658 : 43 54 4f 52 59 0d c6 49  8f
a660 : 4c 45 20 ce 41 4d 45 20  05
a668 : 20 20 20 20 20 20 20 d4  d2
a670 : 59 50 45 20 c2 4c 4b 53  a9
a678 : 20 c1 44 44 52 45 53 53  56
a680 : 20 20 31 53 54 20 c4 41  43
a688 : 54 41 c2 4c 4b 53 20 c3  0e
a690 : 4c 4f 53 45 44 20 d0 52  2f
a698 : 4f 54 45 43 54 20 d2 45  e7
a6a0 : 43 53 20 d2 45 43 cc 54  3a
a6a8 : 48 20 d3 d3 c2 53 0d c6  f8
a6b0 : 49 4c 45 20 ce 41 4d 45  2b
a6b8 : 20 20 20 20 20 20 20 20  b8
a6c0 : 20 20 d4 59 50 45 20 20  41
a6c8 : 20 c2 4c 4f 43 4b 53 20  62
a6d0 : 20 20 28 c1 44 44 52 45  7d
a6d8 : 53 53 29 20 20 20 31 53  92
a6e0 : 54 20 c4 41 54 41 20 c2  f3
a6e8 : 4c 4f 43 4b 53 20 20 20  0d
a6f0 : c3 4c 4f 53 45 44 20 20  4f
a6f8 : 20 d0 52 4f 54 45 43 54  24
a700 : 20 20 20 20 28 d2 45 43  f1
a708 : 4f 52 44 53 20 20 20 d2  25
a710 : 45 43 4f 52 44 20 cc 45  18
a718 : 4e 47 54 48 20 20 20 d3  53
a720 : 49 44 45 20 d3 45 43 54  fe
a728 : 4f 52 20 c2 4c 4f 43 4b  e4
a730 : 53 29 20 0d 0d 20 20 20  54
a738 : 20 20 20 20 20 55 53 45  f9
a740 : 44 20 20 4c 45 46 54 0d  18
a748 : c2 4c 4f 43 4b 53 3a 20  e5
a750 : 20 20 20 20 20 20 20 20  50
a758 : 20 20 0d c2 4c 4f 43 4b  07
a760 : 20 c1 56 41 49 4c 41 42  9f
a768 : 49 4c 49 54 59 20 cd 41  05
a770 : 50 0d 1d 1d 1d 1d 1d 1d  9b
a778 : 1d 1d 1d 1d a3 a3 a3 a3  3c
a780 : a3 a3 a3 b7 b7 b7 b7 b7  5c
a788 : b7 b7 b7 a3 a3 a3 a3 a3  aa
a790 : a3 a3 00 0d d4 4f 54 41  42
a798 : 4c 20 44 49 53 4b 20 43  c5
a7a0 : 41 50 41 43 49 54 59 20  9f
a7a8 : 49 4e 20 42 4c 4f 43 4b  4c
a7b0 : 53 3a 20 36 38 33 0d c2  c6
a7b8 : 4c 4f 43 4b 53 20 55 53  18
a7c0 : 45 44 20 28 41 4c 4c 4f  7b
a7c8 : 43 41 54 45 44 29 20 20  b8
a7d0 : 20 20 20 20 3a 20 20 20  72
a7d8 : 20 20 0d c2 4c 4f 43 4b  87
a7e0 : 53 20 46 52 45 45 20 28  6f
a7e8 : 4c 45 46 54 20 43 41 50  b5
a7f0 : 41 43 49 54 59 29 20 20  4f
a7f8 : 3a 20 20 20 20 20 0d 20  c6
a800 : 20 20 20 20 20 20 20 20  00
a808 : 20 20 20 20 20 20 20 20  08
a810 : 20 20 20 20 20 20 20 20  10
a818 : 20 20 20 31 20 20 20 20  3a
a820 : 20 20 20 20 20 20 20 20  20
a828 : 20 20 20 20 20 20 20 32  4c
a830 : 20 20 20 20 20 20 20 20  30
a838 : 20 20 c2 4c 4f 43 4b 53  85
a840 : 0d d3 45 43 54 4f 52 20  3a
a848 : 20 30 20 31 20 32 20 33  29
a850 : 20 34 20 35 20 36 20 37  dc
a858 : 20 38 20 39 20 30 20 31  2a
a860 : 20 32 20 33 20 34 20 35  96
a868 : 20 36 20 37 20 38 20 39  49
a870 : 20 30 20 20 20 c1 56 41  a0
a878 : 49 4c 41 42 4c 45 20 20  30
a880 : d5 53 45 44 20 20 c6 52  9b
a888 : 45 45 0d d4 52 41 43 4b  21
a890 : 0d 20 20 20 20 20 c8 49  72
a898 : 44 44 45 4e 20 c6 49 4c  0f
a8a0 : 45 53 11 9d 9d 9d 9d 9d  ff
a8a8 : 9d 9d 9d 9d 9d 9d 9d a3  b3
a8b0 : a3 a3 a3 b7 b7 b7 b7 a3  64
a8b8 : a3 a3 a3 00 20 20 20 20  da
a8c0 : 20 20 c4 49 53 4b 20 c1  de
a8c8 : 44 44 52 45 53 53 20 c3  43
a8d0 : 48 41 4e 47 45 0d 1d 1d  a1
a8d8 : 20 20 20 20 20 20 a3 a3  ed
a8e0 : a3 a3 a3 a3 b7 b7 b7 b7  3a
a8e8 : b7 b7 b7 a3 a3 a3 a3 a3  0a
a8f0 : a3 0d 1d 1d 20 d4 55 52  a7
a8f8 : 4e 20 4f 46 46 20 41 4c  f6
a900 : 4c 20 44 52 49 56 45 53  bb
a908 : 20 45 58 43 45 50 54 20  b2
a910 : 54 48 45 20 4f 4e 45 0d  74
a918 : 1d 1d 20 20 20 20 20 20  94
a920 : 20 20 20 20 20 54 4f 20  7f
a928 : 42 45 20 43 48 41 4e 47  d4
a930 : 45 44 0d 11 1d 1d 20 20  78
a938 : 20 20 20 20 20 20 20 20  38
a940 : 20 20 20 20 20 20 cf 4c  57
a948 : 44 3a 20 20 20 38 20 20  3a
a950 : 39 20 20 31 30 20 20 31  af
a958 : 31 0d 1d 1d 20 c4 49 53  ef
a960 : 4b 20 44 52 49 56 45 20  b3
a968 : 44 45 56 49 43 45 0d 1d  da
a970 : 1d 20 20 20 20 20 20 20  6d
a978 : 20 20 20 20 20 20 20 20  78
a980 : 20 ce 45 57 3a 20 20 20  a9
a988 : 38 20 20 39 20 20 31 30  28
a990 : 20 20 31 31 0d 11 1d 1d  3b
a998 : 20 20 20 20 cf cb 20 20  f1
a9a0 : 20 c3 41 4e 43 45 4c 0d  66
a9a8 : 00 cf 4c 44 3a 20 2d 2d  df
a9b0 : 11 9d 9d 9d 9d 9d 9d 9d  23
a9b8 : ce 45 57 3a 20 2d 2d 00  66
a9c0 : 20 c4 49 53 4b 20 d7 52  b9
a9c8 : 49 54 45 20 d0 52 4f 54  16
a9d0 : 45 43 54 0d 1d 1d 1d 1d  d7
a9d8 : 1d 1d 1d 1d 1d 1d 20 a3  f1
a9e0 : a3 a3 a3 a3 b7 b7 b7 b7  3a
a9e8 : b7 b7 b7 b7 a3 a3 a3 a3  8d
a9f0 : a3 0d 1d 1d 1d 1d 1d 1d  6e
a9f8 : 1d 1d 1d 1d 20 d3 54 41  03
aa00 : 54 45 3a 20 20 d2 45 41  ba
aa08 : 44 2f d7 52 49 54 45 11  92
aa10 : 9d 9d 9d 9d 9d 9d 9d 9d  0f
aa18 : 9d 9d 9d 20 d0 52 4f 54  75
aa20 : 45 43 54 20 4f 4e 11 11  ee
aa28 : 9d 9d 9d 9d 9d 9d 9d 9d  27
aa30 : 9d 9d 9d c5 58 49 54 00  dd
aa38 : 20 20 20 20 20 20 20 20  38
aa40 : 20 20 c4 49 53 4b 4e 41  16
aa48 : 4d 45 20 c3 48 41 4e 47  0f
aa50 : 45 0d 1d 1d 20 20 20 20  cb
aa58 : 20 20 20 20 20 20 a3 a3  6d
aa60 : a3 a3 a3 b7 b7 b7 b7 b7  3c
aa68 : a3 a3 a3 a3 a3 0d 1d 1d  8b
aa70 : 20 20 c5 4e 54 45 52 20  d4
aa78 : 4e 45 57 20 4e 41 4d 45  f1
aa80 : 3a 0d 1d 1d 20 28 d2 45  45
aa88 : 54 55 52 4e 20 54 4f 20  07
aa90 : 53 4b 49 50 29 20 a3 a3  4f
aa98 : a3 a3 a3 a3 a3 a3 a3 a3  97
aaa0 : a3 a3 a3 a3 a3 a3 0d 1d  38
aaa8 : 1d 20 20 20 20 20 20 20  a5
aab0 : 20 20 20 20 20 20 20 c9  04
aab8 : c4 3a 11 20 a3 a3 91 91  a2
aac0 : 91 9d 9d 00 20 20 20 d3  b2
aac8 : 59 53 54 45 4d 20 d3 45  38
aad0 : 54 54 49 4e 47 53 0d 1d  e8
aad8 : 1d 20 20 20 a3 a3 a3 a3  3f
aae0 : a3 a3 a3 a3 a3 a3 a3 a3  df
aae8 : a3 a3 a3 0d 1d 1d 20 c4  ac
aaf0 : 49 53 4b 20 20 20 20 20  7e
aaf8 : 20 c4 52 49 56 45 20 ce  e6
ab00 : 4f 3a 20 20 30 20 20 31  5f
ab08 : 0d 1d 1d 20 a3 a3 a3 a3  1c
ab10 : 20 20 20 20 20 c4 45 56  36
ab18 : 49 43 45 20 ce 4f 3a 20  e9
ab20 : 20 38 20 20 39 20 20 31  e0
ab28 : 30 20 20 31 31 0d 1d 1d  c1
ab30 : 20 c1 4e 4e 2e 20 c4 49  18
ab38 : 53 4b 20 c5 58 43 48 41  35
ab40 : 4e 47 45 3a 20 d9 45 53  57
ab48 : 20 20 ce 4f 0d 11 1d 1d  1e
ab50 : 20 d0 52 49 4e 54 45 52  d7
ab58 : 20 20 20 c1 55 54 4f 20  3e
ab60 : cc c6 3a 20 20 d9 45 53  af
ab68 : 20 20 ce 4f 0d 1d 1d 20  a4
ab70 : a3 a3 a3 a3 a3 a3 a3 20  68
ab78 : c4 45 56 49 43 45 20 ce  1a
ab80 : 4f 3a 20 20 20 34 20 20  5d
```

Listing. »XDOS« (Fortsetzung)

```
ab88 : 20 35 0d 1d 1d 20 20 20  bd
ab90 : 20 d3 45 43 55 4e 44 41  af
ab98 : 52 59 20 c1 44 44 52 3a  fb
aba0 : 20 20 3c 20 20 20 20 20  a7
aba8 : 3e 00 0d 1d 1d 20 20 20  61
abb0 : d0 52 49 4e 54 45 52 20  be
abb8 : cd 45 4e 55 0d 1d 1d 20  d5
abc0 : 20 20 a3 a3 a3 a3 b7 b7  f3
abc8 : b7 b7 a3 a3 a3 a3 0d 1d  7e
abd0 : 1d 20 20 c3 48 4f 4f 53  61
abd8 : 45 20 d0 52 49 4e 54 45  8f
abe0 : 52 3a 20 20 20 cd d0 d3  b7
abe8 : 20 38 30 31 2f 38 30 33  32
abf0 : 0d 1d 1d 20 20 20 20 20  9b
abf8 : 20 20 20 20 20 20 20 20  f8
ac00 : 20 20 20 20 20 20 20 c5  4c
ac08 : 50 53 4f 4e 11 9d 9d 9d  4f
ac10 : 9d 9d 9d 20 d5 53 45 52  99
ac18 : 20 20 20 20 20 1d 1d 1d  ee
ac20 : 20 cf cb 11 9d 9d 9d c5  06
ac28 : 58 49 54 0d 1d 1d 20 20  57
ac30 : d3 45 4c 45 43 54 20 c9  4d
ac38 : 4e 54 45 52 46 41 43 45  52
ac40 : 3a 20 ce 4f 4e 45 9d 9d  e9
ac48 : 9d 9d 9d 11 20 c3 45 4e  0f
ac50 : 54 52 4f 4e 49 43 53 11  89
ac58 : 9d 9d 9d 9d 9d 9d 9d 9d  57
ac60 : 9d 9d 9d 20 d5 53 45 52  e9
ac68 : 11 0d 1d 1d 20 20 d0 52  d6
ac70 : 49 4e 54 49 4e 47 20 d1  62
ac78 : 55 41 4c 49 54 59 3a 20  e3
ac80 : cd 41 54 52 49 58 20 20  65
ac88 : ce cc d1 20 11 0d 1d 1d  5d
ac90 : 20 20 d0 52 49 4e 54 20  d7
ac98 : cd 4f 44 45 3a 20 20 20  2c
aca0 : 20 20 20 20 d0 49 43 41  c3
aca8 : 20 20 c5 4c 49 54 45 0d  3a
acb0 : 1d 1d 20 28 d0 49 54 43  98
acb8 : 48 20 d3 45 54 54 49 4e  58
acc0 : 47 29 20 20 20 20 c3 4f  59
acc8 : 4d 50 52 45 53 53 45 44  e8
acd0 : 00 20 20 20 20 20 c4 49  95
acd8 : 53 4b 20 cd 45 4e 55 0d  c9
ace0 : 1d 20 20 20 20 20 a3 a3  f2
ace8 : a3 b7 b7 b7 a3 a3 a3 0d  4c
acf0 : 1d 20 d3 43 52 41 54 43  82
acf8 : 48 20 20 20 20 c6 4f 52  77
ad00 : 4d 41 54 0d 1d 20 d2 45  4d
ad08 : 4e 41 4d 45 20 20 20 20  b7
ad10 : d6 41 4c 49 44 41 54 45  ed
ad18 : 0d 1d 20 20 c3 4f 50 59  6b
ad20 : 20 20 20 20 20 d3 48 4f  bd
ad28 : 57 20 d3 45 51 0d 1d 20  5f
ad30 : c3 4f 4e 43 41 54 20 20  0e
ad38 : 20 20 cc 49 53 54 20 c4  a6
ad40 : 49 52 0d 1d c3 4c 4f 53  1c
ad48 : 45 20 d7 c6 cf 20 20 d0  8c
ad50 : 52 49 4e 54 20 c4 49 52  57
ad58 : 0d 11 1d 20 20 20 20 20  fd
ad60 : 20 20 d2 45 54 55 52 4e  c3
ad68 : 00 20 20 20 20 20 20 c6  96
ad70 : 4f 52 4d 41 54 20 c4 49  50
ad78 : 53 4b 20 20 20 28 d2 45  96
ad80 : 54 55 52 4e 20 54 4f 20  ff
ad88 : 53 4b 49 50 29 0d 1d 1d  87
ad90 : 20 20 20 20 20 20 a3 a3  a5
ad98 : a3 a3 b7 b7 b7 a3 a3 a3  60
ada0 : a3 0d 1d 1d c4 49 53 4b  2f
ada8 : 20 ce 41 4d 45 3a 20 20  10
adb0 : 20 20 20 20 20 20 20 20  b0
adb8 : 20 20 20 20 20 20 20 20  b8
adc0 : 20 c9 c4 3a 0d 1d 1d 20  ac
adc8 : 20 20 20 20 20 20 20 20  c8
add0 : 20 20 a3 a3 a3 a3 a3 a3  8b
add8 : a3 a3 a3 a3 a3 a3 a3 a3  d7
ade0 : a3 a3 20 20 20 20 20 20  25
ade8 : a3 a3 0d 00 20 20 d3 48  83
adf0 : 4f 57 20 d3 45 51 2f d5  b5
adf8 : 53 52 20 c6 49 4c 45 20  a2
ae00 : 20 28 d2 45 54 55 52 4e  67
ae08 : 11 9d 9d 54 4f 11 9d 9d  09
ae10 : 9d 9d 53 4b 49 50 29 0d  90
ae18 : 91 91 1d 1d 20 20 a3 a3  36
ae20 : a3 a3 a3 b7 b7 b7 b7 b7  fc
ae28 : b7 b7 a3 a3 a3 a3 a3 0d  18
ae30 : 1d 1d 20 d4 59 50 45 3a  20
ae38 : 20 20 20 20 d3 45 51 20  61
ae40 : 20 20 d5 53 45 52 0d 1d  a5
ae48 : 1d 20 cf 55 54 50 55 54  da
ae50 : 3a 20 20 41 4c 4c 20 20  b3
ae58 : 46 49 4c 54 45 52 45 44  65
ae60 : 0d 11 1d 1d 20 c6 49 4c  d7
ae68 : 45 3a 11 20 a3 a3 a3 a3  40
ae70 : a3 a3 a3 a3 a3 a3 a3 a3  6f
ae78 : a3 a3 a3 a3 0d 91 91 1d  28
ae80 : 1d 1d 1d 1d 1d 1d 1d 1d  80
ae88 : 00 20 20 2d 20 cd 4f 52  98
ae90 : 45 20 2d 00 14 14 14 14  8b
ae98 : 14 14 14 14 14 14 11 11  86
aea0 : 11 11 11 11 11 11 11 11  a0
aea8 : 11 11 11 11 11 11 11 11  a8
aeb0 : 11 11 11 11 11 91 91 91  b7
aeb8 : 91 91 91 91 91 91 91 91  b7
aec0 : 91 91 91 91 91 91 91 91  bf
aec8 : 91 91 91 91 00 0d 11 90  86
aed0 : 09 0e 08 20 c5 4e 44 20  07
aed8 : 4f 46 20 46 49 4c 45 20  68
aee0 : 20 c2 59 54 45 53 3a 20  5a
aee8 : 00 20 cc 49 53 54 2f d0  8b
aef0 : 52 49 4e 54 20 c4 49 52  f7
aef8 : 45 43 54 4f 52 59 0d 1d  3c
af00 : 20 a3 a3 a3 a3 a3 b7 b7  f5
af08 : b7 b7 b7 b7 b7 b7 b7 b7  07
af10 : a3 a3 a3 a3 a3 0d 1d 20  39
af18 : 20 cc 49 53 54 20 42 59  5d
af20 : 20 54 59 50 45 3a 20 20  b2
af28 : d0 52 47 20 20 d3 45 51  50
af30 : 20 20 d5 53 52 20 20 d2  8c
af38 : 45 4c 0d 11 1d 20 20 28  ad
af40 : d2 45 54 55 52 4e 20 54  35
af48 : 4f 20 53 4b 49 50 29 20  e2
af50 : 20 cc 49 53 54 20 41 4c  77
af58 : 4c 20 46 49 4c 45 53 0d  c5
af60 : 11 11 1d a3 a3 a3 a3 a3  e3
af68 : a3 a3 a3 a3 a3 a3 a3 a3  67
af70 : a3 a3 a3 a3 a3 a3 a3 a3  6f
af78 : a3 a3 a3 a3 a3 a3 a3 a3  77
af80 : a3 a3 a3 a3 a3 a3 a3 a3  7f
af88 : 00 c3 4f 4e 54 11 9d 9d  87
af90 : 9d 9d c1 42 4f 52 54 00  8d
af98 : 20 cf cb 20 11 9d 9d 9d  46
afa0 : 9d 20 20 20 20 20 00 20  9d
afa8 : 20 20 d3 43 52 41 54 43  3d
afb0 : 48 0d 1d 20 20 20 a3 a3  a3
afb8 : b7 b7 b7 a3 a3 00 91 20  6e
afc0 : 20 20 20 20 28 d2 45 54  d3
afc8 : 55 52 4e 20 54 4f 20 53  c5
afd0 : 4b 49 50 29 0d 11 11 1d  d1
afd8 : a3 a3 a3 a3 a3 a3 a3 a3  d7
afe0 : a3 a3 a3 a3 a3 a3 a3 a3  df
afe8 : a3 a3 a3 a3 a3 a3 a3 a3  e7
aff0 : a3 a3 a3 a3 a3 a3 a3 a3  ef
aff8 : a3 a3 a3 a3 a3 00 91 20  c5
b000 : d2 45 4e 41 4d 45 0d 1d  9e
b008 : 1d 20 20 a3 a3 b7 b7 a3  d0
b010 : a3 00 91 20 20 20 28 d2  65
b018 : 45 54 55 52 4e 20 54 4f  fd
b020 : 20 53 4b 49 50 29 0d 11  8a
b028 : 1d 1d 20 20 20 20 20 c6  f1
b030 : 49 4c 45 3a 11 20 a3 a3  20
b038 : a3 a3 a3 a3 a3 a3 a3 a3  37
b040 : a3 a3 a3 a3 a3 a3 0d 1d  d8
b048 : 1d 20 ce 45 57 20 ce 41  06
b050 : 4d 45 3a 11 20 a3 a3 a3  e6
b058 : a3 a3 a3 a3 a3 a3 a3 a3  57
b060 : a3 a3 a3 a3 a3 00 20 20  2d
b068 : c3 4f 50 59 0d 1d 1d 20  81
b070 : 20 a3 b7 b7 a3 00 20 c3  89
b078 : 4f 4e 43 41 54 0d 1d 1d  44
b080 : 20 a3 a3 b7 b7 a3 a3 00  79
b088 : 20 c3 4c 4f 53 45 20 d7  16
b090 : c6 cf 0d 1d 1d 20 a3 a3  ce
b098 : a3 b7 b7 b7 a3 a3 a3 00  e1
b0a0 : 20 20 20 20 20 20 c3 48  7f
b0a8 : 45 43 4b 20 c4 49 53 4b  e0
b0b0 : 11 9d 9d 9d 9d 9d 9d 9d  23
b0b8 : 9d 9d 9d a3 a3 a3 b7 b7  a5
b0c0 : b7 b7 a3 a3 a3 00 20 20  ab
b0c8 : d8 c4 49 52 45 43 54 4f  fd
b0d0 : 52 59 11 9d 9d 9d 9d 9d  3f
b0d8 : 9d 9d 9d 9d 9d a3 a3 a3  2c
b0e0 : b7 b7 b7 b7 a3 a3 a3 20  7e
b0e8 : 00 20 20 d2 45 53 43 52  fb
b0f0 : 41 54 43 48 11 9d 9d 9d  e5
b0f8 : 9d 9d 9d 9d 9d 9d a3 a3  1b
b100 : b7 b7 b7 b7 b7 a3 a3 20  df
b108 : 00 20 20 d3 43 52 41 54  0f
b110 : 43 48 20 d0 52 4f 54 45  15
b118 : 43 54 11 9d 9d 9d 9d 9d  f6
b120 : 9d 9d 9d 9d 9d 9d 9d 9d  1f
b128 : 9d 9d a3 a3 a3 a3 b7 b7  97
b130 : b7 b7 b7 b7 b7 a3 a3 a3  16
b138 : a3 00 0d 11 1d 1d 1d 1d  aa
b140 : 1d 1d 1d 1d 1d 1d 1d 1d  40
b148 : 20 20 20 cf cb 20 20 20  f9
b150 : 20 c3 41 4e 43 45 4c 00  fc
b158 : 91 20 20 28 d2 45 54 55  5a
b160 : 52 4e 20 54 4f 20 53 4b  46
b168 : 49 50 29 0d 11 11 1d 1d  0e
b170 : 1d 1d d4 59 50 45 28 53  f3
b178 : 29 3a 20 20 d3 45 51 20  b7
b180 : 20 d0 52 47 20 20 d5 53  87
b188 : 52 20 20 d2 45 4c 20 20  c4
b190 : c4 45 4c 20 0d 11 1d 1d  16
b198 : 1d 1d c1 54 54 52 49 42  c0
b1a0 : 55 54 45 53 3a 20 20 d7  b0
b1a8 : c6 cf 20 20 20 20 c3 4c  0d
b1b0 : 4f 53 45 44 11 9d 9d 9d  32
b1b8 : 9d 9d 9d 9d 9d 9d 9d 9d  b7
b1c0 : 9d 9d 9d 20 d2 2f d7 20  dd
b1c8 : 20 20 d0 52 4f 54 45 43  aa
b1d0 : 54 45 44 0d 11 1d 1d 1d  22
b1d8 : 1d 20 20 20 ce 41 4d 45  c8
b1e0 : 3a 20 20 11 a3 a3 a3 a3  82
b1e8 : a3 a3 a3 a3 a3 a3 a3 a3  e7
b1f0 : a3 a3 a3 a3 91 20 20 20  9d
b1f8 : cf cb 00 c4 49 53 4b 20  e2
b200 : c4 41 4d 41 47 45 20 d2  a5
b208 : 45 50 4f 52 54 0d c4 49  e7
b210 : 53 4b 20 53 55 43 43 45  82
b218 : 53 53 46 55 4c 4c 59 20  1e
b220 : 52 45 41 44 2e 0d 2d 2d  48
b228 : 20 c5 52 52 4f 52 53 20  1f
b230 : 44 45 54 45 43 54 45 44  49
b238 : 2e 0d 20 20 20 d0 45 52  3b
b240 : 46 4f 52 4d 41 4e 43 45  8a
b248 : 20 d4 45 53 54 11 9d 9d  0e
b250 : 9d 9d 9d 9d 9d 9d 9d 9d  4f
b258 : 9d 9d 9d 9d 9d 9d a3 a3  7b
b260 : a3 a3 a3 b7 b7 b7 b7 b7  3c
b268 : b7 a3 a3 a3 a3 a3 00 20  e6
b270 : 20 20 20 20 20 20 20 20  70
b278 : 20 20 2a 2a 2a 2a 20 c3  74
b280 : c1 d5 d4 c9 cf ce 20 2a  e2
b288 : 2a 2a 2a 11 9d 9d 9d 9d  ed
b290 : 9d 9d 9d 9d 9d 9d 9d 9d  8f
b298 : a3 a3 a3 a3 a3 a3 a3 11  72
b2a0 : 9d 9d 9d 9d 9d 9d 9d 9d  9f
b2a8 : 9d 9d 9d 9d 9d 9d 9d 9d  a7
b2b0 : 9d 9d 9d 9d 9d 9d 9d c1  f8
b2b8 : 4c 4c 20 44 41 54 41 20  b7
b2c0 : 4f 4e 20 49 4e 53 45 52  a1
b2c8 : 54 45 44 20 44 49 53 4b  46
b2d0 : 20 57 49 4c 4c 20 42 45  d1
b2d8 : 20 4c 4f 53 54 00 c4 49  48
b2e0 : 53 4b 20 ce 41 4d 45 3a  c3
b2e8 : 20 2d 2d 2d 2d 2d 2d 2d  db
b2f0 : 2d 2d 2d 2d 2d 2d 2d 2d  f0
b2f8 : 2d 20 20 c9 c4 3a 20 2d  70
b300 : 2d 20 2d 2d 20 20 20 c4  3b
b308 : 4f 53 20 c6 4f 52 4d 41  21
b310 : 54 3a 20 d6 c9 c3 20 31  02
b318 : 35 34 31 2c c3 c2 cd 20  03
b320 : 32 30 33 31 2f 34 30 34  1b
b328 : 30 c3 c2 cd 20 38 30 35  93
b330 : 30 2f 38 32 35 30 c8 49  d7
b338 : 44 44 45 4e 20 c6 49 4c  af
b340 : 45 53 0d c5 4d 50 54 59  86
b348 : 20 c4 49 53 4b 2e 0d d4  8b
b350 : 52 41 43 4b 20 58 58 20  e4
b358 : 20 d3 45 43 54 4f 52 20  65
b360 : 58 58 20 20 20 20 20 20  b4
b368 : 0d 20 20 20 20 50 4f 54  fc
b370 : 45 4e 54 49 41 4c 20 46  9e
b378 : 49 4c 45 53 2e 0d 00 00  ee
b380 : 00 00 00 00 00 00 00 00  81
b388 : 00 00 00 00 00 00 00 00  89
b390 : 00 01 04 07 0a 0d 10 02  40
b398 : 05 08 0b 0e 11 03 06 09  79
b3a0 : 0c 0f 12 02 22 42 62 82  bb
b3a8 : a2 c2 e2 80 40 20 10 08  c9
b3b0 : 04 02 01 55 31 3a 32 20  8e
b3b8 : 30 20 54 54 20 53 53 0d  9c
b3c0 : 42 2d 50 3a 32 20 30 0d  f3
b3c8 : 23 49 4e 30 3a 58 44 4f  40
b3d0 : 53 20 54 45 53 54 44 49  6d
b3d8 : 53 4b 2c 50 57 53 30 3a  2b
b3e0 : d8 c4 4f 53 20 d0 52 2f  89
b3e8 : c4 53 4b 2e 44 41 54 41  11
b3f0 : 58 30 3a 20 20 20 20 20  b7
b3f8 : 20 20 20 20 20 20 20 20  f8
b400 : 20 20 20 20 20 20 20 20  00
b408 : 20 20 20 20 20 20 20 20  08
b410 : 20 20 20 20 20 20 20 20  10
b418 : 0d 4d 2d 57 01 01 01 41  a1
b420 : 4d 2d 57 77 00 02 29 49  10
b428 : 13 0d 52 d5 0d 20 44 45  7f
b430 : 4c 20 53 45 51 20 50 52  06
b438 : 47 20 55 53 52 20 52 45  49
b440 : 4c c5 52 52 4f 52 20 20  96
b448 : 20 20 20 20 20 20 20 20  48
b450 : 20 20 20 20 20 20 20 20  50
b458 : 20 20 20 20 20 d4 52 2f  e5
b460 : d3 43 20 20 20 20 20 20  a5
b468 : d7 f7 17 37 57 77 97 b7  e7
b470 : b9 b9 ba ba ba ba ba ba  ee
```

Listing. »XDOS« (Schluß)

Der unentdeckte Floppy-Fehler

Im DOS der Commodore-Laufwerke 1541, 1570 und 1571 hat sich ein bösartiger Fehler eingeschlichen, der sich wie das Ungeheuer von Loch Ness verhält: 99 Prozent aller Programmierer wird er nie begegnen, doch dann schlägt er zu.

Das Speichern einer veränderten Datei bei gleichzeitigem Ersetzen der alten Datei (Save and Replace) durch Angabe des Klammeraffen »@« zu Beginn des Filenamens ist seit langem als fehlerhaft entlarvt worden. Doch ein weiterer Fehler liegt in der Behandlung relativer Dateien und wurde bislang in keiner Fachzeitschrift veröffentlicht. Dies liegt vor allem daran, daß er nur in speziellen Situationen auftritt, welche sich trotz aller Forschungsbemühungen noch nicht exakt definieren ließen.

Der Fehler besteht darin, daß bei der unmittelbaren Abfolge von Lese- und Schreibzugriffen auf ein und denselben Sektor einer Diskette die Floppy »ins Leere« schreibt. Dies hat nicht nur die Zerstörung der Datei selbst, sondern auch unkontrollierte Nebenwirkungen auf andere Teile der Diskette zur Folge. Entscheidend ist hierbei die Verwendung des Befehls »B-P« (Buffer-Pointer); offensichtlich wird dieser nämlich in der gegebenen Situation fehlinterpretiert.

Was unternimmt man nun gegen diesen sehr seltenen, aber verheerenden Fehler?

Dafür gibt es zwei Lösungen: Zum einen kann man zwischen Lese- und Schreibzugriff die Floppy initialisieren (Diskettenbefehl »I«); dies verlangsamt jedoch die Arbeit bis ins Unerträgliche.

Eine völlig irrationale, aber funktionierende Lösung besteht nun – man glaubt es kaum – im doppelten Ausführen des Befehls »B-P«, also der wiederholten Positionierung des Buffer-Pointers; dann treten keinerlei Probleme mehr auf, und der Geschwindigkeitsverlust hält sich in Grenzen.

Wenn nun ein Befehl wie B-P doppelt anzugeben ist, sei die Frage gestellt, ob die Floppy denn »schwerhörig« ist ...

Commodore ist wenigstens teilweise in Schutz zu nehmen: Im Handbuch zur 1571 findet sich, recht gut versteckt, ein ähnlicher Hinweis, aus welchem jedoch niemand schlau wurde. Mitgeteilt wurde uns dieser DOS-Fehler von Said Baloui, einem bekannten Autor von Artikeln, Büchern und Programmen. Bei der Entwicklung seiner Dateiverwaltung »Master-Base« traf er auf dieses Phänomen.

(Florian Müller/sk)

1571-Power im C 64-Modus

Obwohl der C128 und die Floppy 1571 über schnelle Laderoutinen verfügen, bleibt im C64-Modus alles beim alten. Dieses Hilfsprogramm beseitigt dieses Manko und bringt auch das Laden von Programmen im C64-Modus auf Trab.

Welcher Besitzer von C 128 und Floppy 1571 hat sich darüber nicht schon geärgert: Da besitzt man eine Gerätekonfiguration, die im C 128-Modus zeigt, wie schnell Ladevorgänge sein können, doch wenn man in den C 64-Modus umschaltet, ist davon nichts mehr zu merken. Die 1571 simuliert dann die alte und bekannt langsame Floppy 1541 und braucht beispielsweise zum Laden eines 200 Block langen Programmes 140 Sekunden. Abhilfe bieten verschiedene Floppy-Speeder, was in der Regel jedoch mit nicht unerheblichen Kosten und Umbaumaßnahmen verbunden ist. Das hier vorgestellte Programm »128-64« geht einen anderen Weg. Es wird als Vorspann vor ein beliebiges normales C 64-Programm gehängt, und man erhält nun ein schnelladefähiges Programm, das im C 128-Modus gestartet wird. Die Erstellung eines solchen schnelladefähigen C 128-Programmes aus einem C 64-Programm beinhaltet zwar mehrere Schritte, ist aber im Grunde recht einfach:

1. Schritt: Zunächst lädt man im C 128-Modus das Programm »128-64 BAS« (Listing 1).
2. Schritt: Über ⟨F8⟩ oder durch den Befehl »POKE 46,29 : NEW« wird der Monitor des C 128 aufgerufen.
3. Schritt: Mit der Ladefunktion des Monitors wird jetzt das zu behandelnde C 64-Programm dazu geladen:

```
L "NAME", 08, 1D01.
```

4. Schritt: Jetzt muß das Ende des geladenen C 64-Programms ermittelt werden, indem man »M 00AE« eingibt. Die Inhalte beziehungsweise Werte der jetzt angezeigten Speicherstellen $00AE und $00AF muß man sich merken: In $00AE steht das Low-Byte und in $00AF das High-Byte der Adresse des Programmendes.
5. Schritt: Das gesamte Programm wird nun gespeichert mit

```
S "NAME 128", 08, 1C01, HHLL
```

oder

```
POKE 46,28 : SAVE "BASICPROGRAMM 128", 8
```

Dabei bedeutet HH = High-Byte und LL = Low-Byte der Programmende-Adresse. Das ist schon alles! Für das Laden von 200 Blöcken werden statt 200 Sekunden jetzt nur noch zirka 16 Sekunden benötigt. Außerdem können alle so veränderten Programme bedienerfreundlich im C 128-Modus mit RUN "Name*" gestartet werden. Das im Directory an erster Stelle befindliche Programm sogar nur mit ⟨SHIFT RUN/STOP⟩. Grundsätzlich können alle Programmarten auf diese Weise bearbeitet werden. Bei Maschinenprogrammen mit Basic-Start, die mit einem SYS-Befehl beginnen, muß das Vorspannprogramm »128-64 SYS« (Listing 2) verwendet werden. Nach dem Laden (1.Schritt) müssen dabei die Basic-Zeiger des C 64-Modus (für X = PEEK (45) und Y = PEEK (46)) über POKE-Befehle gesetzt werden:

`POKE 7293,X` und `POKE 7297,Y`.

Bei Maschinenprogrammen, egal in welchem Bereich sie laufen, müssen die Werte $1C40, $1C41, $1C54, $1C63 und $1C64 im Programm »128-64 $C0« (Listing 3) geändert werden. Bei mehrteiligen Programmen muß der Teil $1C3D bis $1C64 entsprechend oft vervielfacht werden (Monitorbefehl »T«). Interessierte und Assembler-Programmierer finden in Listing 4 bis 6 die Quellcodes der drei Programme.

(Reiner Moll/Dr. Rudolf Egg)

```
Name : 128-64 bas                   1c01 1d04
---------------------------------------------
1c01 : 0b 1c c3 07 9e 37 31 38   c5
1c09 : 31 00 00 00 a9 f7 8d 05   d5
1c11 : d5 a2 ff 78 9a d8 8e 16   1d
1c19 : d0 20 a3 fd 20 50 fd 20   5f
1c21 : 15 fd 20 5b ff 58 ea ea   ed
1c29 : ea a2 00 bd 3d 1c 9d 34   b0
1c31 : 03 e8 e0 b0 d0 f5 4c 34   4d
1c39 : 03 ea ea ea a2 00 bd 01   ec
1c41 : 1d 9d 01 08 e8 e0 00 d0   a5
1c49 : f5 ee 38 03 ee 3b 03 ad   54
1c51 : 3b 03 c9 b0 d0 e6 ea 20   c6
1c59 : 53 e4 20 bf e3 ad 01 08   de
1c61 : 8d f5 03 ad 02 08 8d f6   e4
1c69 : 03 20 22 e4 ad f5 03 8d   53
1c71 : 01 08 ad f6 03 8d 02 08   75
1c79 : ea ea ea a9 ff 85 14 85   50
1c81 : 15 20 13 a6 18 a5 5f 69   3f
1c89 : 02 85 2d a5 60 69 00 85   aa
1c91 : 2e 20 60 a6 ea 4c ae a7   d7
1c99 : 00 00 00 00 00 00 00 00   9a
1ca1 : 00 00 00 00 00 00 00 00   a2
1ca9 : 00 00 00 00 00 00 00 00   aa
1cb1 : 00 00 00 00 00 00 00 00   b2
1cb9 : 00 00 00 00 00 00 00 00   ba
1cc1 : 00 00 00 00 00 00 00 00   c2
1cc9 : 00 00 00 00 00 00 00 00   ca
1cd1 : 00 00 00 00 00 00 00 00   d2
1cd9 : 00 00 00 00 00 00 00 00   da
1ce1 : 00 00 00 00 00 00 00 00   e2
1ce9 : 00 00 00 00 00 00 00 00   ea
1cf1 : 00 00 00 00 00 00 00 00   f2
1cf9 : 00 00 00 00 00 00 00 00   fa
1d01 : 00 00 00 88 ff 88 f7 88   48
```

Listing 1. »128-64 BAS«, bitte mit dem MSE auf Seite 158 im C64-Modus eingeben.

```
Name : 128-64 sys                   1c01 1d04
---------------------------------------------
1c01 : 0b 1c c3 07 9e 37 31 38   c5
1c09 : 31 00 00 00 a9 f7 8d 05   d5
1c11 : d5 a2 ff 78 9a d8 8e 16   1d
1c19 : d0 20 a3 fd 20 50 fd 20   5f
1c21 : 15 fd 20 5b ff 58 ea ea   ed
1c29 : ea a2 00 bd 3d 1c 9d 34   b0
1c31 : 03 e8 e0 b0 d0 f5 4c 34   4d
1c39 : 03 ea ea ea a2 00 bd 01   ec
1c41 : 1d 9d 01 08 e8 e0 00 d0   a5
1c49 : f5 ee 38 03 ee 3b 03 ad   54
1c51 : 3b 03 c9 b0 d0 e6 ea 20   c6
1c59 : 53 e4 20 bf e3 ad 01 08   de
1c61 : 8d f5 03 ad 02 08 8d f6   e4
1c69 : 03 20 22 e4 ad f5 03 8d   53
1c71 : 01 08 ad f6 03 8d 02 08   75
1c79 : ea ea ea a9 03 85 2d a9   2c
1c81 : 08 85 2e 20 60 a6 ea 4c   5b
1c89 : ae a7 00 00 00 00 00 00   0b
1c91 : 00 00 00 00 00 00 00 00   92
1c99 : 00 00 00 00 00 00 00 00   9a
1ca1 : 00 00 00 00 00 00 bc a0   d6
1ca9 : 00 00 00 00 00 00 00 00   aa
1cb1 : 00 00 00 00 00 00 00 00   b2
1cb9 : 00 00 00 00 00 00 00 00   ba
1cc1 : 00 00 00 00 00 00 00 00   c2
1cc9 : 00 00 00 00 00 00 00 00   ca
1cd1 : 00 00 00 00 00 00 00 00   d2
1cd9 : 00 00 00 00 00 00 00 00   da
1ce1 : 00 00 00 00 00 00 00 00   e2
1ce9 : 00 00 00 00 00 00 00 00   ea
1cf1 : 00 00 00 00 00 00 00 00   f2
1cf9 : 00 00 00 00 00 00 00 00   fa
1d01 : 00 00 00 88 ff 88 f7 88   48
```

Listing 2. »128-64 SYS«, bitte mit dem MSE auf Seite 158 im C64-Modus eingeben.

```
Name : 128-64 $c0                   1c01 1d04
---------------------------------------------
1c01 : 0b 1c c3 07 9e 37 31 38   c5
1c09 : 31 00 00 00 a9 f7 8d 05   d5
1c11 : d5 a2 ff 78 9a d8 8e 16   1d
1c19 : d0 20 a3 fd 20 50 fd 20   5f
1c21 : 15 fd 20 5b ff 58 ea ea   ed
1c29 : ea a2 00 bd 3d 1c 9d 34   b0
1c31 : 03 e8 e0 b0 d0 f5 4c 34   4d
1c39 : 03 ea ea ea a2 00 bd 01   ec
1c41 : 1d 9d 00 c0 e8 e0 00 d0   7c
1c49 : f5 ee 38 03 ee 3b 03 ad   54
1c51 : 3b 03 c9 d0 d0 e6 20 53   06
1c59 : e4 20 bf e3 20 22 e4 ea   36
1c61 : ea 4c 00 c0 ea 00 00 00   38
1c69 : 00 00 00 00 00 00 00 00   6a
1c71 : 00 00 00 00 00 00 00 00   72
1c79 : 00 00 00 00 00 00 00 00   7a
1c81 : 00 00 00 00 00 00 00 00   82
1c89 : 00 00 00 00 00 00 00 00   8a
1c91 : 00 00 00 00 00 00 00 00   92
1c99 : 00 00 00 00 00 00 00 00   9a
1ca1 : 00 00 00 00 00 00 00 00   a2
1ca9 : 00 00 00 00 00 00 00 00   aa
1cb1 : 00 00 00 00 00 00 00 00   b2
1cb9 : 00 00 00 00 00 00 00 00   ba
1cc1 : 00 00 00 00 00 00 00 00   c2
1cc9 : 00 00 00 00 00 00 00 00   ca
1cd1 : 00 00 00 00 00 00 00 00   d2
1cd9 : 00 00 00 00 00 00 00 00   da
1ce1 : 00 00 00 00 00 00 00 00   e2
1ce9 : 00 00 00 00 00 00 00 00   ea
1cf1 : 00 00 00 00 00 00 00 00   f2
1cf9 : 00 00 00 00 00 00 00 00   fa
1d01 : 00 00 00 88 ff 88 f7 88   48
```

Listing 3. »128-64 $C0, bitte mit dem MSE auf Seite 158 im C64-Modus eingeben.

```
01C0D  A9 F7     LDA #$F7      Umschaltung in den 64'er Modus
01C0F  8D 05 D5  STA $D505
01C12  A2 FF     LDX #$FF
01C14  78        SEI
01C15  9A        TXS
01C16  D8        CLD
01C17  8E 16 D0  STX $D016
01C1A  20 A3 FD  JSR $FDA3
01C1D  20 50 FD  JSR $FD50
01C20  20 15 FD  JSR $FD15
01C23  20 5B FF  JSR $FF5B
01C26  58        CLI
01C27  EA        NOP
01C28  EA        NOP
01C29  EA        NOP
01C2A  A2 00     LDX #$00      Rette die Peek/Poke-Schleife vor dem Überschreiben
                                             in den Kassettenpuffer.
01C2C  BD 3D 1C  LDA $1C3D,X
01C2F  9D 34 03  STA $0334,X
01C32  E8        INX
01C33  E0 B0     CPX #$B0
01C35  D0 F5     BNE $1C2C
01C37  4C 34 03  JMP $0334     Springe in den Kassettenpuffer und mache da weiter.
01C3A  EA        NOP
01C3B  EA        NOP
01C3C  EA        NOP
01C3D  A2 00     LDX #$00      Lade das Programm an den Basicanfang.
01C3F  BD 01 1D  LDA $1D01,X
01C42  9D 01 08  STA $0801,X
01C45  E8        INX
01C46  E0 00     CPX #$00
01C48  D0 F5     BNE $1C3F
01C4A  EE 38 03  INC $0338
01C4D  EE 3B 03  INC $033B
01C50  AD 3B 03  LDA $033B
01C53  C9 B0     CMP #$B0
01C55  D0 E6     BNE $1C3D
01C57  EA        NOP
01C58  20 53 E4  JSR $E453     BASIC-Vektoren laden.
01C5B  20 BF E3  JSR $E3BF     RAM für BASIC initialisieren.
01C5E  AD 01 08  LDA $0801     BASIC-Anfang merken.
01C61  8D F5 03  STA $03F5
01C64  AD 02 08  LDA $0802
01C67  8D F6 03  STA $03F6
01C6A  20 22 E4  JSR $E422     Zeiger auf Programmstart und NEW.
01C6D  AD F5 03  LDA $03F5     BASIC-Anfang wieder herstellen.
01C70  8D 01 08  STA $0801
01C73  AD F6 03  LDA $03F6
01C76  8D 02 08  STA $0802
01C79  EA        NOP
01C7A  EA        NOP
01C7B  EA        NOP
01C7C  A9 FF     LDA #$FF      BASIC-Zeiger wieder herstellen.
01C7E  85 14     STA $14
01C80  85 15     STA $15
01C82  20 13 A6  JSR $A613
01C85  18        CLC
01C86  A5 5F     LDA $5F
01C88  69 02     ADC #$02
01C8A  85 2D     STA $2D
01C8C  A5 60     LDA $60
01C8E  69 00     ADC #$00
01C90  85 2E     STA $2E
01C92  20 60 A6  JSR $A660     CLR
01C95  EA        NOP
01C96  4C AE A7  JMP $A7AE     Programmstart.
```

Listing 4. Der Source-Code von »128-64 BAS«.


```
Erster Teil wie 128-64 BAS.

01C79  EA        NOP
01C7A  EA        NOP
01C7B  EA        NOP
01C7C  A9 03     LDA #$03      In $1c7D muß der Wert von (PEEK(45)) gepokt werden.
01C7E  85 2D     STA $2D
01C80  A9 08     LDA #$08      In $1c81 muß der Wert von (PEEK(46)) gepokt werden.
01C82  85 2E     STA $2E
01C84  20 60 A6  JSR $A660     BASIC-Zeiger anpassen (CLR).
01C87  EA        NOP
01C88  4C AE A7  JMP $A7AE     Programmstart.
```

Listing 5. Der Source-Code von »128-64 SYS«.

```
Beispiel für den Bereich $C000 bis $CFFF.
Erster Teil wie 128-64 BAS.

01C3A  EA        NOP
01C3B  EA        NOP
01C3C  EA        NOP
01C3D  A2 00     LDX #$00      Lade ab $1d01 nach $C000.
01C3F  BD 01 1D  LDA $1D01,X
01C42  9D 00 C0  STA $C000,X
01C45  E8        INX
01C46  E0 00     CPX #$00
01C48  D0 F5     BNE $1C3F
01C4A  EE 38 03  INC $0338
01C4D  EE 3B 03  INC $033B
01C50  AD 3B 03  LDA $033B
01C53  C9 D0     CMP #$D0      Ist High-Byte schon D0 ?
01C55  D0 E6     BNE $1C3D
01C57  20 53 E4  JSR $E453
01C5A  20 BF E3  JSR $E3BF
01C5D  20 22 E4  JSR $E422     NEW
01C60  EA        NOP
01C61  EA        NOP
01C62  4C 00 C0  JMP $C000     Springe zur neuen Adresse und starte das Programm.
```

Listing 6. Der Source-Code von »128-64 $C0«.

Floppy 1541: 759 Blocks free

Sie haben richtig gelesen: »759 Blocks free« und das mit der Floppy 1541. »Ultraformat« nutzt die Spuren 36 bis 41 – mit eigenem Directory. Bespielte Disketten können Sie sogar nachträglich um diese Spuren erweitern.

Wie Sie sicher wissen, ist die Floppy 1541 eigentlich ein 41-Track-Laufwerk. Das DOS hingegen nutzt nur die Spuren (Tracks) 1 bis 35, das heißt, sechs Spuren bleiben ungenutzt. Durch geschickte Programmierung werden auch diese letzten sechs Tracks voll genutzt.

Zwei Disketten auf einer Seite

»Ultraformat« formatiert die Tracks 36 bis 41 und richtet auf Track 36 ein Directory ein. So hat man im »oberen« Bereich der Diskette, Track 36 bis 41, genau 95 Blöcke frei. Im »unteren« Bereich, Track 1 bis 35, stehen die gewohnten 664 freien Blöcke zur Verfügung. Da man zwischen diesen Bereichen umschalten kann, hat man quasi zwei Disketten auf einer Seite mit insgesamt 759 freien Blöcken.

Doch genug der Theorie; geben Sie Listing 1, »Ultraformat«, mit dem Checksummer ein. Nach dem Start mit »RUN« bestimmt man mit den Cursortasten, ob eine ganze Diskette (Track 1 bis 41) oder nur Track 36 bis 41 formatiert wird. So lassen sich Disketten, die bereits Informationen enthalten, nachträglich auf 41 Spuren formatieren, ohne die bisherigen Informationen zu zerstören. Soll eine ganze

```
0 "RALPH GAUER /PLZ" OBEN
1    "DISKSWITCH C000"   PRG
1    "DISKSWITCH 033C"   PRG
25   "ULTRAFORMAT"       PRG
19   "ART ULTRAFORMAT"   PRG
2    "SWITCH U-O.BAS"    PRG
1    "SWITCH O-U.BAS"    PRG
55 BLOCKS FREE.
█
```

Bild 1. So sieht das Directory von Spur 36 bis 41 aus.

Diskette formatiert werden, sind Diskettenname und ID einzugeben. Beim Nachformatieren entfallen diese Eingaben.

Nachdem das Programm die gewünschte Formatierung durchgeführt hat, generiert es für die Spuren 36 bis 41 ein eigenes Directory, das auf Spur 36 zu finden ist. Der Diskettenname wird im zweiten Directory übernommen. Hier hat man 95 Blöcke frei zur Verfügung. Anstatt der ID und der Kennung »2a« findet man in diesem Directory die Bezeichnung »oben«.

Danach schreibt das Programm selbständig (ohne Informationsverlust) zwei Versionen eines Spurschaltprogrammes in den »normalen« Bereich der Diskette.

Das Spurschaltprogramm (»Diskswitch c000« oder »Diskswitch 033c«) ist notwendig, um den »oberen« Bereich der Diskette zu nutzen. Es ist ein absolut zu ladendes Programm, das mit »SYS Anfangsadresse« aufgerufen wird, wobei die Anfangsadresse je nach Version dezimal 49152 oder 828 ist. Bei Aufruf des Spurschaltprogrammes wird der NMI-Vektor auf die Umschaltroutine gerichtet. Nun schaltet ein einfaches Drücken von ⟨RESTORE⟩ in den jeweils anderen Bereich der Diskette. Aus Sicherheitsgründen wird beim ersten Betätigen von ⟨RESTORE⟩ immer in den »unteren« Bereich geschaltet, also auf Track 1 bis 35.

Leichter Zugriff auf den neuen Bereich

Auf den oberen Bereich greift man wie gewohnt zu, beispielsweise mit den Befehlen LOAD, SAVE oder OPEN. Mit Kopierprogrammen, die den NMI nicht beeinflussen, kann man sogar von dem »unteren« in den »oberen« Bereich und umgekehrt kopieren.

Wie schon gesagt, bilden die Spuren 36 bis 41 eine eigene Diskette mit einer Kapazität von 95 Blöcken. Dieser Bereich darf unter keinen Umständen validiert werden, da die VALIDATE-Routine des DOS versuchen würde, 664 Blöcke freizugeben (wir haben aber nur 95). Beim nächsten Schreibversuch versucht das Laufwerk dann, auf nicht vorhandene Spuren zugegriffen, was zum Verlust von Informationen führen könnte. In Bild 1 sehen Sie übrigens ein »oberes« Directory.

Sollten Sie einmal den Computer beziehungsweise Computer und Floppy ausschalten, obwohl der Schreib-Lese-Kopf noch auf den Spuren 36 bis 41 ist; und haben Sie kein Spurschaltprogramm auf einer Diskette im oberen Bereich, so läßt sich die Floppy durch einen INITIALIZE-Befehl ohne Diskette im Laufwerk wieder in den Normalmodus zurückversetzen.

Die Spurumschaltung ist auch von Basic möglich. Bei »switch u-o.bas« (Listing 2) wird vom unteren in den oberen Bereich geschaltet. Das Listing 3, »switch o-u.bas«, bewirkt genau das Gegenteil. Diese zwei Listings können Sie in Ihre eigenen Programme einbinden.

(Ralph Gauer/Thomas Lipp)

```
0 REM                                       <062>
100 REM"{RETURN,CTRL-H,GRAPHIC}             <047>
102 REM"U********************************I  <028>
103 REM"-{9SPACE}WRITTEN BY{10SPACE}-       <254>
104 REM"-RALPH GAUER{18SPACE}-              <218>
105 REM"-WEILERSTR. 8{17SPACE}-             <064>
106 REM"-7968 WILFERTSWEILER{10SPACE}-      <106>
107 REM"Q********************************W  <037>
108 REM"-DATUM: 17. JUNI 1987{9SPACE}-      <187>
109 REM"J********************************K  <122>
110 PRINT"{CLR,WHITE,CTRL-H}"CHR$(142):POK
    E 53280,0:POKE 53281,0                  <237>
112 REM                                   **
    ************ ENTSCHEIDUNG ************* <142>
114 GG$(0)="{HOME}GANZE DISKETTE FORMATIER
    EN{RVOFF}"                              <071>
116 GG$(1)="{HOME,DOWN}NUR SPUR 36-41 FORM
    ATIEREN{RVOFF}"                         <004>
118 GG=0:PRINT GG$(0);GG$(1)                <225>
120 PRINT"{HOME,5DOWN}AUSWAHL MIT CURSORTA
    STEN UND [{RVSON}RETURN{RVOFF}]"        <113>
122 PRINT"{RVSON}"GG$(GG)                   <039>
124 WAIT 198,1:GET GG$                      <162>
126 IF GG$=CHR$(13)THEN 136                 <147>
```

Listing 1. »Ultraformat« formatiert auf 41 Spuren. Bitte mit dem Checksummer (Seite 158) eingeben.

```
128 IF GG$="{DOWN}"THEN IF GG=0 THEN PRINT
    GG$(GG):GG=1:GOTO 122                   <236>
130 IF GG$="{UP}"THEN IF GG=1 THEN PRINT G
    G$(GG):GG=0:GOTO 122                    <206>
132 GOTO 124                                <220>
134 REM                                   **
    *********** INITIALISIERUNG ********** <110>
136 OPEN 1,8,15,"UI"                        <146>
138 DEF FN HI(X)=INT(X/256):DEF FN LO(X)=I
    NT(X-FN HI(X)*256)                      <098>
140 ON GG GOTO 200                          <133>
142 REM                                   **
    *********** NORMAL FORMATIEREN ******* <058>
144 I$="{HOME,5DOWN}":PRINT I$"{40SPACE}"; <204>
146 II$=I$+"DISKETTENNAME: ":PRINT II$"{RV
    SON,16SPACE,RVOFF}"                     <216>
148 PRINT II$"±";:NA$=""                    <173>
150 WAIT 198,1:GET O$:POKE 198,0            <078>
152 IF O$=CHR$(13)AND NA$<>""THEN 164       <035>
154 IF O$=CHR$(20)OR O$="{LEFT}"THEN 146    <224>
156 IF ASC(O$)>31 AND ASC(O$)<128 AND LEN(
    NA$)<16 THEN NA$=NA$+O$                 <119>
158 IF ASC(O$)>159 AND LEN(NA$)<16 THEN NA
    $=NA$+O$                                <226>
160 PRINT II$"{RVSON}"NA$"{RVOFF}";:IF LEN
    (NA$)<16 THEN PRINT"±";                 <067>
162 GOTO 150                                <170>
164 PRINT II$NA$"{15SPACE}":PRINT II$"{DOW
    N}TTTTTTTTTTTTTTTT"                     <074>
166 I$="{HOME,7DOWN}":PRINT I$"{40SPACE}"; <211>
168 II$=I$+"DISKETTEN-ID: ":PRINT II$"{RVS
    ON,5SPACE,RVOFF}"                       <001>
170 PRINT II$"±";:DI$=""                    <243>
172 WAIT 198,1:GET O$:POKE 198,0            <100>
174 IF O$=CHR$(13)AND DI$<>""THEN DI$=DI$+
    "{4SPACE}":DI$=LEFT$(DI$,5):GOTO 186    <085>
176 IF O$=CHR$(20)OR O$="{LEFT}"THEN 168    <087>
178 IF ASC(O$)>31 AND ASC(O$)<128 AND LEN(
    DI$)<5 THEN DI$=DI$+O$                  <210>
180 IF ASC(O$)>159 AND LEN(DI$)<5 THEN DI$
    =DI$+O$                                 <156>
182 PRINT II$"{RVSON}"DI$"{RVOFF}";:IF LEN
    (DI$)<5 THEN PRINT"±";                  <041>
184 GOTO 172                                <032>
186 PRINT II$DI$:PRINT II$"{DOWN}TTTTT":ID
    $=LEFT$(DI$,2)                          <042>
188 PRINT#1,"N:"+NA$+","+ID$                <076>
190 OPEN 2,8,2,"#2"                         <105>
192 PRINT#1,"U1"2;0;18;0                    <069>
194 PRINT#1,"B-P"2;162:PRINT#2,DI$;         <068>
196 PRINT#1,"U2"2;0;18;0:CLOSE 2            <081>
198 REM                                   **
    ********** VORBEREITUNG ************** <105>
200 PRINT#1,"M-W"CHR$(81)CHR$(0)CHR$(1)CHR
    $(255):PRINT#1,"I"                      <248>
202 PRINT#1,"M-W"CHR$(8)CHR$(0)CHR$(2)CHR$
    (36)CHR$(0)                             <206>
204 REM                                   **
    ********** UEBERTRAGUNG ************** <113>
206 FOR X=1024 TO 1024+32:READ A:PRINT#1,"
    M-W"CHR$(FN LO(X))CHR$(FN HI(X))CHR$(1
    )CHR$ (A)                               <154>
208 NEXT                                    <218>
210 REM                                   **
    ********** FORMATIERE **************** <075>
212 PRINT#1,"M-W"CHR$(1)CHR$(0)CHR$(1)CHR$
    (224)                                   <180>
214 REM                                   **
    ********** WARTEN ******************** <202>
216 PRINT#1,"M-R"CHR$(1)CHR$(0)CHR$(1):GET
    #1,A$:IF ASC(A$+CHR$(0))>127 THEN 216   <188>
218 REM                                   **
    ********** FORMAT-DATAS ************** <232>
220 DATA 165, 81,201,255,240, 24,165, 81,2
    01, 1,208, 6,169, 18,133, 81,133, 34    <168>
222 DATA 165, 81,201, 25,208, 6,169, 36,1
    33, 81,133, 34, 76,199,250              <084>
224 REM                                   **
    ********** TWO BUMPS ***************** <112>
226 FOR BU=1 TO 2:PRINT#1,"M-W"CHR$(0)CHR$
    (0)CHR$(1)CHR$(192)                     <190>
228 PRINT#1,"M-R"CHR$(0)CHR$(0)CHR$(1):GET
    #1,A$:IF ASC(A$+CHR$(0))>127 THEN 228   <059>
230 NEXT                                    <240>
232 REM                                   **
    ********** SPURSCHALTPROGRAMME ******* <009>
234 GOTO 254                                <122>
236 PRINT#1,"M-W"CHR$(6)CHR$(0)CHR$(2)CHR$
    (1)CHR$(0)                              <136>
238 PRINT#1,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$
    (128)                                   <205>
240 PRINT#1,"M-R"CHR$(0)CHR$(0)CHR$(1):GET
    #1,A$:IF ASC(A$+CHR$(0))>127 THEN 240   <064>
242 RETURN                                  <044>
244 GOSUB 236:PRINT#1,"M-W"CHR$(6)CHR$(0)C
    HR$(2)CHR$(36)CHR$(0)                   <242>
246 PRINT#1,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$
    (176)                                   <215>
248 PRINT#1,"M-R"CHR$(0)CHR$(0)CHR$(1):GET
    #1,A$:IF ASC(A$+CHR$(0))>127 THEN 248   <080>
250 RETURN                                  <052>
252 REM                                   **
    ********** DIRECTORY GENERATOR ******* <007>
254 GOSUB 236:OPEN 2,8,2,"#2"               <029>
256 PRINT#1,"U1"2;0;18;0                    <135>
258 FOR X=4 TO 71:PRINT#1,"B-P"2;X          <183>
260 PRINT#2,CHR$(0);:NEXT                   <112>
262 PRINT#1,"B-P"2;72:PRINT#2,CHR$(17)CHR$
    (252)CHR$(255)CHR$(7);:                 <159>
264 FOR X=76 TO 95 STEP 4:PRINT#1,"B-P"2;X <002>
266 PRINT#2,CHR$(19)CHR$(255)CHR$(255)CHR$
    (7);:NEXT                               <099>
268 FOR X=96 TO 143:PRINT#1,"B-P"2;X        <150>
270 PRINT#2,CHR$(0);:NEXT                   <122>
272 PRINT#1,"B-P"2;162:PRINT#2,"OBEN ";     <233>
274 GOSUB 244:PRINT#1,"U2"2;0;18;0          <173>
276 GOSUB 236                               <214>
278 REM                                   **
    ********** TEXT GENERIEREN *********** <097>
280 PRINT#1,"U1"2;0;18;1                    <160>
282 GOSUB 244                               <172>
284 FOR X=0 TO 255:PRINT#1,"B-P"2;X:PRINT#
    2,CHR$(0);:NEXT                         <226>
286 PRINT#1,"B-P";2;0                       <001>
288 PRINT#2,CHR$(0)CHR$(255)CHR$(192)CHR$(
    0)CHR$(0)"NICHT VALIDIEREN";            <066>
290 PRINT#1,"U2"2;0;18;1                    <043>
292 CLOSE 2:GOSUB 236:PRINT#1,"I"           <176>
294 REM                                   **
    ********** DISKSWITCH C000 *********** <068>
296 OPEN 2,8,2,"DISKSWITCH C000,P,W"        <004>
298 INPUT#1,A,B$,C,D:IF A<>0 THEN CLOSE 2:
    CLOSE 1:PRINT A;B$;C;D:END              <110>
300 PRINT#2,CHR$(FN LO(49152))CHR$(FN HI(4
    9152));                                 <231>
302 FOR X=0 TO 195:READ A:PRINT#2,CHR$(A);
    :NEXT                                   <120>
304 CLOSE 2                                 <069>
306 INPUT#1,A,B$,C,D:IF A<>0 THEN CLOSE 2:
    CLOSE 1:PRINT A;B$;C;D:END              <118>
308 REM                                   **
    ********** DISKSWITCH 033C *********** <129>
310 OPEN 2,8,2,"DISKSWITCH 033C,P,W"        <138>
312 INPUT#1,A,B$,C,D:IF A<>0 THEN CLOSE 2:
    CLOSE 1:PRINT A;B$;C;D:END              <124>
314 PRINT#2,CHR$(FN LO(828))CHR$(FN HI(828
    ));                                     <206>
316 FOR X=0 TO 195:READ A:PRINT#2,CHR$(A);
    :NEXT                                   <134>
318 CLOSE 2                                 <083>
320 REM                                   **
    ********** ENDE ********************** <199>
322 INPUT#1,A,B$,C,D:PRINT A;B$;C;D         <068>
324 CLOSE 1                                 <081>
326 REM VON 49152 BIS 49347               **
    ********** DS C000 DATA ************** <045>
328 DATA 169, 11,141, 24, 3,169,192,141,
    25, 3, 96,120,169,161,141, 24, 3        <249>
330 DATA 169, 2,133,183,169, 20,133,184,1
    69, 15,133,185,169, 8,133,186,169       <223>
332 DATA 193,133,187,169,192,133,188, 32,1
    92,255,169, 20, 32,195,255,173,162      <200>
334 DATA 192,240, 11, 32,109,192,169, 0,1
    41,162,192, 76,142,192, 32, 76,192      <218>
336 DATA 169, 1,141,162,192, 76,142,192,1
    69, 9,133,183,169,163,133,187, 32       <021>
338 DATA 192,255,169, 20, 32,195,255,169,
     8,133,183,169,171,133,187, 32,192      <021>
340 DATA 255,169, 20, 32,195,255, 96,169,
     9,133,183,169,178,133,187, 32,192      <090>
```

```
342 DATA 255,169, 20, 32,195,255,169,  8,1
    33,183,169,186,133,187, 32,192,255     <178>
344 DATA 169, 20, 32,195,255, 96,169,  2,1
    33,183,169,193,133,187, 32,192,255     <037>
346 DATA 169, 20, 32,195,255, 32,  0,192,
    64,  1, 77, 45, 87,  6,  0,  2, 36     <017>
348 DATA  0, 77, 45, 87,  0,  0,  1,176, 7
    7, 45, 87,  6,  0,  2,  1,  0, 77      <061>
350 DATA 45, 87,  0,  0,  1,128, 73, 68, 7
    8                                      <190>
352 REM VON 828 BIS 1023                 **
    ********** DS 033C DATA ************** <010>
354 DATA 169, 71,141, 24,  3,169,  3,141,
    25,  3, 96,120,169,221,141, 24,  3     <238>
356 DATA 169,  2,133,183,169, 20,133,184,1
    69, 15,133,185,169,  8,133,186,169     <249>
358 DATA 253,133,187,169,  3,133,188, 32,1
    92,255,169, 20, 32,195,255,173,222     <205>
360 DATA  3,240, 11, 32,169,  3,169,  0,14
    1,222,  3, 76,202,  3, 32,136,  3      <022>
362 DATA 169,  1,141,222,  3, 76,202,  3,1
    69,  9,133,183,169,223,133,187, 32     <254>
364 DATA 192,255,169, 20, 32,195,255,169,
     8,133,183,169,231,133,187, 32,192     <172>
366 DATA 255,169, 20, 32,195,255, 96,169,
     9,133,183,169,238,133,187, 32,192     <178>
368 DATA 255,169, 20, 32,195,255,169,  8,1
    33,183,169,246,133,187, 32,192,255     <148>
370 DATA 169, 20, 32,195,255, 96,169,  2,1
    33,183,169,253,133,187, 32,192,255     <035>
372 DATA 169, 20, 32,195,255, 32, 60,  3,
    64,  1, 77, 45, 87,  6,  0,  2, 36     <247>
374 DATA  0, 77, 45, 87,  0,  0,  1,176, 7
    7, 45, 87,  6,  0,  2,  1,  0, 77      <087>
376 DATA 45, 87,  0,  0,  1,128, 73, 68,
    0                                      <234>
```

Listing 1. (Schluß)

```
0 REM                                          <062>
136 OPEN 1,8,15,"UI"                           <146>
232 REM *** SPURSCHALTPROGRAMME ***            <230>
236 PRINT#1,"M-W"CHR$(6)CHR$(0)CHR$(2)CHR$
    (1)CHR$(0)                                 <136>
238 PRINT#1,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$
    (128)                                      <205>
240 PRINT#1,"M-R"CHR$(0)CHR$(0)CHR$(1):GET
    #1,A$:IF ASC(A$+CHR$(0))>127 THEN 240      <064>
244 PRINT#1,"M-W"CHR$(6)CHR$(0)CHR$(2)CHR$
    (36)CHR$(0)                                <216>
246 PRINT#1,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$
    (176)                                      <215>
248 PRINT#1,"M-R"CHR$(0)CHR$(0)CHR$(1):GET
    #1,A$:IF ASC(A$+CHR$(0))>127 THEN 248      <080>
250 CLOSE 1:END                                <184>
```

Listing 2. »switch u-o.bas«, bitte mit dem Checksummer (Seite 158) eingeben.

```
0 REM                                          <062>
136 OPEN 1,8,15,"UI"                           <146>
232 REM *** SPURSCHALTPROGRAMME ***            <230>
236 PRINT#1,"M-W"CHR$(6)CHR$(0)CHR$(2)CHR$
    (1)CHR$(0)                                 <136>
238 PRINT#1,"M-W"CHR$(0)CHR$(0)CHR$(1)CHR$
    (128)                                      <205>
240 PRINT#1,"M-R"CHR$(0)CHR$(0)CHR$(1):GET
    #1,A$:IF ASC(A$+CHR$(0))>127 THEN 240      <064>
250 CLOSE 1:END                                <184>
```

Listing 3. »switch o-u.bas«, bitte mit dem Checksummer (Seite 158) eingeben.

Sprites manipulieren – direkt auf Diskette

Selbstentworfene Sprites und Zeichensätze sind das »i-Tüpfelchen«, das eigenen Programmen die individuelle Note verleiht. Mit »Perfect View« können Sie direkt auf der Diskette nach Sprites und Zeichensätzen suchen und diese nach Ihren Vorstellungen ändern oder auch »klauen«.

»Perfect View« (Listing 1) wird mit »LOAD "PERFECT VIEW",8« von Diskette geladen und mit RUN gestartet. Sie sehen nun das Titelbild, von dem Sie durch Tastendruck ins Hauptprogramm kommen. In der linken oberen Ecke wird eine Sprite-Matrix dargestellt. Am rechten Rand befinden sich zwei Kolonnen mit Nullen. Hier wird später der Blockinhalt in Zahlendarstellung ausgegeben. Dabei dient die linke Spalte zur Anzeige der Dezimal- und die rechte zur Anzeige der Hexadezimalzahlen. Links daneben liegt die Zeichenmatrix. In der Linie aus Punkten am unteren Rand wird später der Inhalt in Form von ASCII-Zeichen ausgegeben.

Drücken Sie nun bitte die Taste <Cursor-rechts>. Das veranlaßt den Computer, Block 18/0 zu lesen. Unser »Cursor« zeigt nun auf Track 18, Sektor 0, Byte 0. Das erkennen Sie aus den Zahlen bei den Symbolen, die in der rechten unteren Ecke senkrecht angeordnet stehen.

↑: Byte   X: Sektor   o: Track

Nun stehen Ihnen folgende Funktionen zur Verfügung:

<Cursor/Joystick-rechts>

Sie kommen ein Byte weiter. Wenn Sie am Ende des Blocks angelangt sind, wird automatisch der logisch nächste Block geladen. Sind Sie am Ende des Files, so wird Block 18/0 geladen (dies war zu Beginn der Fall).

<Cursor/Joystick-links>

Die Anzeige wird um ein Byte zurückgeschaltet. Dabei ist es sehr oft interessant, von einem späteren Block einer Datei wieder rückwärts zu suchen. Es gibt allerdings keinen Zeiger, der auf den vorigen Block zeigt. Dafür hat Perfect View einen jeweils 256 Byte langen Stack eingerichtet, auf dem alle Zeiger abgelegt werden.

Sie werden oftmals auch einen längeren Programmabschnitt bearbeiten wollen. Dabei ist es lästig, wenn bei jedem Verlassen des einen Blocks der nächste erst geladen wird. Deshalb behält Perfect View immer die letzten vier geladenen Blöcke im Speicher. Sie können also innerhalb eines Bereichs von vier Blöcken bequem arbeiten, ohne daß dazwischen auf das Laufwerk zugegriffen wird.

<Cursor/Joystick-aufwärts/abwärts>

Bei der schnellen Suche nach Sprites wird bei jedem Schritt um 3 Byte gesprungen.

Nun wäre es interessant, auch ein anderes File als das Directory zu durchleuchten. Dazu suchen Sie im Inhaltsverzeichnis nach dem Beginn des Filenamens und stellen den

Cursor auf das zweite Byte davor. Nun werfen Sie einen Blick auf die Zahlenanzeige. Merken Sie sich bitte die Nummer, auf die der Cursor zeigt, und die unmittelbar darunterstehende. Diese geben den ersten Track und Sektor des gewählten Programms an. Jetzt wählen Sie folgende Funktion an:

<**X**>: Sektor und Track einstellen

Dieses Kommando veranlaßt den Computer, auf den Anfang des letzten Blocks im Puffer zu gehen. Daraufhin wird die Anzeige des aktuellen Sektors (»X«) Null. Jetzt können Sie den gewünschten Sektor (dezimal) eingeben und die Eingabe mit <RETURN> abschließen. Wenn Sie nur <RETURN> drücken, wird der alte Wert beibehalten. Nun geben Sie auf gleiche Weise den Track ein. Daraufhin wird der gewählte Block geladen.

Perfect View hat vier Modi, die durch Abkürzungen in der rechten unteren Ecke des Bildschirms symbolisiert werden:

VIW: Durchsuchen des Programms
TXT: Texteditor
FIG: Ändern der Zahlen
SPR: Sprite-Editor
CHR: Zeicheneditor

Im Augenblick befinden Sie sich im Suchmodus. Hier haben Sie folgende weitere Funktionen, um die angezeigten Farben zu ändern:

<**T**> : Textfarbe
<**H**> : Hintergrundfarbe
<**R**> : Rahmenfarbe
<**F**> : Spritefarbe
<**G**> : Rasterfarbe
<**M**> : Multicolorfarbe 0
<**N**> : Multicolorfarbe 1
<**A**> : Auflösung des Sprite

Mit dieser Taste wird die Darstellung des Sprite zwischen Hires- und Multicolor-Modus hin- und hergeschaltet.

<↑>/<←>: Vertikale/Horizontale Spriteausdehnung

Damit können die beiden Sprites in der Mitte des Bildschirms in X-und Y-Richtung gedehnt beziehungsweise gestaucht werden.

<**$**>: Directory

zeigt das Inhaltsverzeichnis der eingelegten Diskette.

<**B**>: BAM Display

Es wird Block 18/0 geladen. Dort befindet sich die BAM (Block Availability Map), die angibt, welche Blöcke auf der Diskette belegt sind. Dabei stellen die Spalten die Spuren und die Zeilen die Sektoren dar. Ein Kreis signalisiert einen freien Block, ein reverses Blank einen belegten.

<**E**>: Fehlerkanal lesen

<**C**>: Floppybefehl senden

Die Anzeige der ASCII-Zeichen wird gelöscht, und es erscheint »C>«. Geben Sie nun einen beliebigen Befehl an die Diskettenstation ein (zum Beispiel »I« für »Initialize«).

<**o**>: Bildschirm-Codes anzeigen

Viele Programme verwenden keine ASCII-Codes, sondern speichern Texte in Form von Bildschirm-Codes. Mit dieser Funktion wird die Anzeige auf Bildschirm-Codes umgeschaltet.

<**F7**>: Texteditor

Es ist die erste Funktion, mit der Sie Änderungen des Disketteninhalts vornehmen können. Es empfiehlt sich, Änderungen nur auf einer Kopie zu machen. Sie können damit Textstellen überschreiben. Durch erneuten Druck von <F7> wird dieser Modus wieder verlassen.

Achtung: Änderungen werden immer erst dann gespeichert, wenn Sie den aktuellen Block verlassen.

<**F5**>: Dezimalzahlen editieren

Hier können Sie das Byte, auf das der Cursor gerade zeigt, ändern. Die Eingabe muß mit <RETURN> abgeschlossen werden, woraufhin der Cursor um eine Position weitergesetzt wird. Mit <F5> kommen Sie wieder in den Suchmodus. Wenn Sie diesen Modus statt mit <F5> mit <F6> anwählen, braucht die Eingabe nicht mit <RETURN> abgeschlossen werden. Der Rücksprungbefehl ist wieder <F5>.

<**STOP**>: Block neu laden

Nach einer ungewollten Änderung ist es möglich, mit diesem Befehl den aktuellen Block neu in den Speicher zu laden.

<**+**>: Anfang des nächsten Blocks

Der Cursor springt an den Anfang des nächsten Blocks.

<**–**>: Ende des vorhergehenden Blocks

<**Q**>: Quit

Wenn Sie Perfect View verlassen, so sollten Sie dies immer auf diesem Weg tun, anstatt den Computer einfach auszuschalten. Falls Sie eine Änderung gemacht haben, wird diese nämlich zuvor gespeichert.

<**S**>: Zustand speichern

Damit steht Ihnen eine sehr praktische Funktion zur Verfügung. Es wird nämlich der gesamte Zustand gespeichert, inklusive aller Farben, der vier Puffer und des Stacks. Sie sollten diese Möglichkeit immer nutzen, wenn Sie eine interessante Programmstelle gefunden haben. Wenn Sie später diesen Zustand wieder laden, so können Sie genau an dieser Stelle weiterarbeiten, um sich zum Beispiel wieder ein Sprite aus diesem Programm zu holen.

<**L**>: Zustand laden

Damit wird ein gespeicherter Zustand wieder geladen.

<**F1**>: Sprite-Editor

Damit sind wir bei der wohl wichtigsten Einrichtung von Perfect View angelangt, dem Sprite-Editor. Wundern Sie sich bitte nicht, wenn beim Anwählen dieser Funktion das Sprite verschoben wird. Es kann nämlich vorkommen, daß sich ein Sprite am Übergang zwischen zwei Blöcken befindet. Dabei stören aber die ersten beiden Bytes des einen Blocks. Diese bilden nämlich einen Zeiger auf den nächsten Block. Darum werden sie im Sprite-Editor-Modus bei der Anzeige übersprungen.

Der Sprite-Editor verfügt über folgende Funktionen:

<**S**>: Sprite speichern

Damit wird das Sprite als Programmfile gespeichert. Die Ladeadresse des File ist 13 * 64. Dieses Sprite können Sie dann in einem eigenen Programm verwenden oder auch mit einem anderen Editor bearbeiten und später wieder an dieser Stelle einfügen. Sie können sich auf diese Weise eine umfangreiche Spritesammlung anlegen, auf die Sie dann bei eigenen Programmen zurückgreifen können.

<**L**>: Sprite laden

Damit ist es möglich, ein eigenes Sprite, oder eines, das Sie aus einem anderen Programm »geklaut« haben, an der gerade bearbeiteten Stelle einzufügen.

Weitere Steuerkommandos:

Srollen in vier Richtungen:
Joystick oder Cursor betätigen

<**X**> : Spiegeln um X-Achse
<**Y**> : Spiegeln um Y-Achse
<**I**> : Invertieren
<**CLR**> : Löschen
<**STOP**> : Original zurückholen
<**A**> : Auflösung ändern
<**1**> : Punkt löschen
<**2**> : Punkt in Spritefarbe setzen
<**3**> : Punkt in Multicolorfarbe 0 setzen
<**4**> : Punkt in Multicolorfarbe 1 setzen
<**$**> : Directory anzeigen
<**E**> : Fehlerkanal lesen
<**C**> : Floppybefehl senden

<T>, <H>, <R>,
<F>, <G>, <M>, <N>: Farbe ändern
<F1>: Sprite-Editor beenden
<F2>: Zeicheneditor. Hier können Sie den Zeichensatz eines Programms editieren. Sie haben folgende Möglichkeiten:

<X> : Spiegeln um X-Achse
<Y> : Spiegeln um Y-Achse
Cursortasten: Scrollen
<,> : Nach links drehen
<.> : Nach rechts drehen
<CLR> : Löschen
<I> : Invertieren
<STOP> : Original zurückholen
<T>, <H>, <R>,
<F>, <G>, <M>, <N>: Farben ändern

Mit einem derart leistungsfähigen Programm wie Perfect View haben Sie also ein optimales Werkzeug in der Hand, welches Ihnen das Editieren und Ändern von Sprites und Zeichensätzen gestattet. (Th. Führinger/sk)

```
Name : perfect view      0801 2156
------------------------------------
0801 : 18 08 c3 07 9e 20 32 31   05
0809 : 32 39 20 50 45 52 46 45   75
0811 : 43 54 20 56 49 45 57 00   6d
0819 : 00 00 20 56 4f 4e 20 54   7d
0821 : 48 4f 4d 41 53 20 46 55   86
0829 : 45 48 52 49 4e 47 45 52   29
0831 : 2c 48 4f 45 4c 4c 45 20   7a
0839 : 31 30 2c 41 2d 32 38 30   5b
0841 : 33 20 53 43 48 57 41 52   ab
0849 : 5a 45 4e 42 41 43 48 92   96
0851 : 4c a2 1f 0b 02 01 06 00   58
0859 : 00 00 00 07 10 03 ff 00   53
0861 : 00 00 00 00 00 00 00 00   62
0869 : 00 00 00 00 a9 ff 8d c0   bc
0871 : 02 a9 e8 85 f9 a9 d7 85   8a
0879 : fa a9 01 8d c1 02 a9 10   2d
0881 : 18 65 f9 85 f9 a5 fa 69   06
0889 : 00 85 fa a9 01 8d c2 02   cb
0891 : ee c0 02 a9 08 18 65 f9   60
0899 : 85 f9 a5 fa 69 00 85 fa   86
08a1 : ad c5 02 c9 08 f0 36 a0   0d
08a9 : 00 8c c3 02 ac c0 02 b9   6d
08b1 : 40 03 ac c3 02 c0 06 f0   36
08b9 : 06 4a c8 c0 06 d0 fa 29   54
08c1 : 03 a8 b9 bc 02 ac c3 02   b7
08c9 : 91 f9 c8 91 f9 ee c3 02   e5
08d1 : ee c3 02 ad c3 02 c9 08   5b
08d9 : d0 d2 4c 0b 09 a9 00 8d   80
08e1 : c3 02 a9 80 ae c3 02 ac   8a
08e9 : c0 02 bd 26 09 39 40 03   40
08f1 : d0 06 ad bc 02 4c fc 08   4e
08f9 : ad be 02 ac c3 02 91 f9   a2
0901 : ee c3 02 ad c3 02 c9 08   8b
0909 : d0 d8 ee c2 02 ad c2 02   f6
0911 : c9 04 f0 03 4c 91 08 ee   c8
0919 : c1 02 ad c1 02 c9 16 f0   28
0921 : 03 4c 7f 08 60 80 40 20   77
0929 : 10 08 04 02 01 a9 ff 85   e7
0931 : f7 a9 03 85 f8 a2 15 a0   a9
0939 : 00 c8 a9 cf 91 f7 ad c5   1d
0941 : 02 c9 10 d0 05 c8 a9 f7   73
0949 : 91 f7 c0 18 d0 eb a9 28   6c
0951 : 18 65 f7 85 f7 a5 f8 69   2e
0959 : 00 85 f8 ca d0 d9 60 01   13
0961 : ad 60 09 f0 26 a0 00 b9   7a
0969 : 50 03 c9 20 90 12 c9 a0   b3
0971 : f0 09 c9 60 b0 0a 29 3f   e3
0979 : 4c 83 09 a9 20 4c 83 09   83
0981 : a9 2e 99 70 07 c8 c0 1f   ae
0989 : d0 dd 60 a2 00 bd 50 03   e9
0991 : 9d 70 07 e8 e0 1f d0 f5   7b
0999 : 60 a9 22 85 fa a9 04 85   1f
09a1 : fb a2 00 8e c0 02 bd 56   7f
09a9 : 03 20 e5 09 a0 00 ad ca   ad
09b1 : 02 91 fa c8 ad c9 02 91   a8
09b9 : fa c8 ad c8 02 91 fa c8   c6
09c1 : c8 ad cd 02 91 fa c8 ad   83
09c9 : cc 02 91 fa a9 28 18 65   61
09d1 : fa 85 fa a9 00 65 fb 85   a8
09d9 : fb ee c0 02 ae c0 02 e0   76
09e1 : 13 d0 c3 60 8d c2 02 29   a2
09e9 : 0f aa bd 3e 0a 8d cc 02   c9
09f1 : ad c2 02 4a 4a 4a 4a aa   3f
09f9 : bd 3e 0a 8d cd 02 a9 30   fe
0a01 : 8d c8 02 8d c9 02 8d ca   9d
0a09 : 02 ae c2 02 8a c9 c8 90   8e
0a11 : 06 ee ca 02 38 e9 64 c9   79
0a19 : 64 90 06 ee ca 02 38 e9   96
0a21 : 64 aa 4c 39 0a ca ee c8   59
0a29 : 02 ad c8 02 c9 3a d0 08   36
0a31 : a9 30 8d c8 02 ee c9 02   31
0a39 : e0 00 d0 e9 60 30 31 32   3b
0a41 : 33 34 35 36 37 38 39 01   bf
0a49 : 02 03 04 05 06 a9 19 85   8c
0a51 : fa a9 d8 85 fb a2 00 a0   1d
0a59 : 00 b9 26 09 3d 5c 03 d0   45
0a61 : 06 ad bc 02 4c 6b 0a ad   51
0a69 : be 02 91 fa c8 c0 08 d0   40
0a71 : e8 a0 00 a9 28 18 65 fa   ad
0a79 : 85 fa a9 00 65 fb 85 fb   2a
0a81 : e8 e0 08 d0 d2 60 a9 18   fc
0a89 : 85 fa a9 04 85 fb a2 08   49
0a91 : a0 08 a9 cf 91 fa 88 d0   4e
0a99 : f9 a9 28 18 65 fa 85 fa   ae
0aa1 : a9 00 65 fb 85 fb ca d0   28
0aa9 : e7 60 a9 42 85 fa a9 07   58
0ab1 : 85 fb a0 00 ad c7 02 20   bd
0ab9 : c8 0a ad ce 02 20 c8 0a   24
0ac1 : ad cf 02 20 c8 0a 60 20   79
0ac9 : e5 09 a0 00 ad ca 02 91   b7
0ad1 : fa c8 ad c9 02 91 fa c8   fe
0ad9 : ad c8 02 91 fa c8 c8 ad   11
0ae1 : cd 02 91 fa c8 ad cc 02   a4
0ae9 : 91 fa a5 fa 18 69 28 85   38
0af1 : fa 60 78 20 4e 0a 20 61   b6
0af9 : 09 20 9a 09 20 ab 0a 20   a2
0b01 : 6d 08 58 60 a0 ff 91 fa   db
0b09 : 88 d0 fb 60 a9 ef 2d 11   f5
0b11 : d0 8d 11 d0 a5 80 8d 8a   b0
0b19 : 02 a2 00 a9 08 20 d2 ff   6e
0b21 : a9 92 20 d2 ff 8a 18 69   fd
0b29 : 7a 85 fb a9 00 85 fa a8   03
0b31 : 91 fa 88 d0 fb e8 e0 06   12
0b39 : d0 eb a0 78 84 fb a9 01   07
0b41 : 8d 1d d0 20 05 0b e6 fb   d1
0b49 : a9 12 20 05 0b a2 40 bd   e6
0b51 : 63 16 9d 7f 03 ca d0 f7   d0
0b59 : a2 17 bd 53 08 9d bb 02   c1
0b61 : ca d0 f7 a9 00 8d 97 12   b5
0b69 : a9 0e 8d fa 07 20 41 0e   6f
0b71 : 20 e3 0d a9 03 8d 15 d0   8e
0b79 : 8d 17 d0 8d 1d d0 [illegible] e8   48
0b81 : 8d 00 d0 8d 02 d0 a9 79   34
0b89 : 8d 01 d0 a9 ad 8d 03 d0   f5
0b91 : a9 00 8d 1b d0 8d 21 d0   a1
0b99 : 8d 10 d0 a9 02 8d 1c d0   36
0ba1 : 20 2b 0e a9 06 8d 20 d0   ff
0ba9 : a9 0d 8d f8 07 8d f9 07   2e
0bb1 : ad 11 d0 09 10 8d 11 d0   8f
0bb9 : ad 00 dc aa e0 7f f0 03   c6
0bc1 : 4c 81 0c a5 c6 f0 f1 20   82
0bc9 : d1 10 aa a0 00 20 d8 0b   db
0bd1 : a5 ff d0 e4 4c 6b 0c a0   d8
0bd9 : 00 c9 52 d0 0e a9 20 85   26
0be1 : fa a9 d0 85 fb 20 21 0e   f6
0be9 : 4c 66 0c e0 48 d0 0e a9   1e
0bf1 : 21 85 fa a9 d0 85 fb 20   32
0bf9 : 21 0e 4c 66 0c a9 bc 85   0d
0c01 : fa a9 02 85 fb e0 46 d0   83
0c09 : 17 a0 02 20 21 0e ad be   ac
0c11 : 02 8d 28 d0 8d 27 d0 8d   6e
0c19 : 29 d0 20 f3 0a 4c 66 0c   e6
0c21 : e0 4d d0 0e a0 01 20 21   73
0c29 : 0e 20 6d 08 20 2b 0e 4c   d0
0c31 : 66 0c e0 4e d0 0e a0 03   a5
0c39 : 20 21 0e 20 6d 08 20 2b   5f
0c41 : 0e 4c 66 0c e0 47 d0 09   2e
0c49 : 20 21 0e 20 f3 0a 4c 66   0f
0c51 : 0c e0 54 d0 0b a0 08 20   13
0c59 : 21 0e 20 e3 0d 4c 66 0c   eb
0c61 : a9 00 85 ff 60 a9 01 85   ce
0c69 : ff 60 e0 41 d0 25 a9 18   06
0c71 : 38 ed c5 02 8d c5 02 20   a1
0c79 : 2e 09 20 6d 08 4c b9 0b   c1
0c81 : 29 01 f0 1d 8a 29 02 f0   e6
0c89 : 31 8a 29 04 f0 1d 8a 29   3f
0c91 : 08 f0 04 e0 1d d0 06 20   df
0c99 : b6 0f 4c b9 0b e0 91 d0   c1
0ca1 : 06 20 42 10 4c b9 0b e0   ca
0ca9 : 9d d0 0b a9 01 8d d4 02   7a
0cb1 : 20 4f 10 4c b9 0b e0 11   a0
0cb9 : d0 0b a9 03 8d d4 02 20   a1
0cc1 : 4f 10 4c b9 0b e0 58 d0   1d
0cc9 : 06 20 f6 10 4c b9 0b e0   1f
0cd1 : 5e d0 08 a9 03 4d 17 d0   67
0cd9 : 8d 17 d0 e0 5f d0 08 a9   32
0ce1 : 03 4d 1d d0 8d 1d d0 e0   b3
0ce9 : 45 d0 06 20 8d 11 4c b9   22
0cf1 : 0b e0 24 d0 06 20 f6 11   ef
0cf9 : 4c b9 0b e0 2b d0 16 20   d3
0d01 : b6 0f ad c7 02 f0 f1 a9   66
0d09 : ff 8d c7 02 20 41 0e 20   86
0d11 : b6 0f 4c b9 0b e0 2d d0   a7
0d19 : 21 a9 01 8d d4 02 20 4f   7d
0d21 : 10 ae c7 02 e8 f0 d1 a9   6b
0d29 : 00 8d c7 02 20 41 0e a9   ba
0d31 : 01 8d d4 02 20 4f 10 4c   c4
0d39 : b9 0b e0 88 d0 06 20 cc   18
0d41 : 12 4c b9 0b e0 87 d0 0b   ed
0d49 : a9 00 8d 73 13 20 1d 13   91
0d51 : 4c b9 0b e0 8b d0 0b a9   17
0d59 : 01 8d 73 13 20 1d 13 4c   30
0d61 : b9 0b e0 43 d0 06 20 2a   52
0d69 : 14 4c b9 0b e0 42 d0 06   e2
0d71 : 20 85 15 4c b9 0b e0 4f   39
0d79 : d0 0b ad 60 09 49 01 8d   40
0d81 : 60 09 20 61 09 e0 86 d0   ed
0d89 : 06 20 a6 16 4c b9 0b e0   8c
0d91 : 85 d0 06 20 28 1a 4c b9   fc
0d99 : 0b e0 03 d0 14 20 33 0f   1c
0da1 : 20 60 0e 20 41 0e 20 f3   66
0da9 : 0a a9 00 8d cb 02 4c b9   ab
0db1 : 0b e0 4c d0 0c 20 1e 1f   d2
0db9 : 20 41 0e 20 f3 0a 4c b9   36
0dc1 : 0b e0 53 d0 06 20 bf 1e   c8
0dc9 : 4c b9 0b e0 51 d0 10 20   ed
0dd1 : d1 10 c9 51 d0 09 8d 05   dc
0dd9 : 80 20 da 10 4c e2 fc 4c   8a
0de1 : b9 0b ad c4 02 8d 86 02   ce
0de9 : 20 44 e5 a9 43 8d 8d 05   bb
0df1 : a9 1e 8d 45 07 a9 56 8d   e8
0df9 : 6d 07 a9 6b 8d 90 04 a9   82
0e01 : 73 8d 99 04 a9 72 8d 57   35
0e09 : 07 a9 57 8d 95 07 a9 71   87
0e11 : 8d a7 07 20 2e 09 20 a7   33
0e19 : 12 20 87 0a 20 f3 0a 60   e9
0e21 : 18 b1 fa 69 01 29 0f 91   b6
0e29 : fa 60 ad be 02 8d 27 d0   61
0e31 : 8d 28 d0 ad bd 02 8d 25   28
0e39 : d0 ad bf 02 8d 26 d0 60   1e
0e41 : 20 10 0f a0 00 b1 fa 99   ee
0e49 : 40 03 c8 c0 3f d0 f6 60   6c
0e51 : 23 55 31 3a 20 35 20 30   3f
0e59 : 20 30 30 20 30 30 0d a9   ae
0e61 : 31 8d 53 0e a9 05 a2 08   4d
0e69 : a8 20 ba ff a9 01 a2 51   a0
0e71 : a0 0e 20 bd ff 20 c0 ff   dc
0e79 : a9 0f a2 08 a8 20 ba ff   ca
0e81 : a9 0e a2 52 a0 0e 20 bd   9b
0e89 : ff 20 c0 ff a2 05 20 c6   29
0e91 : ff a0 00 84 fa a9 7b 18   8c
0e99 : 6d c6 02 85 fb 20 cf ff   9b
0ea1 : 91 fa c8 d0 f8 a9 0f 20   55
0ea9 : c3 ff a9 05 20 c3 ff 20   d7
0eb1 : cc ff 60 a9 32 8d 53 0e   c3
0eb9 : a9 05 a2 08 a8 20 ba ff   05
0ec1 : a9 01 a2 51 a0 0e 20 bd   34
0ec9 : ff 20 c0 ff a2 05 20 c9   6f
0ed1 : ff a9 7b 18 6d c6 02 85   a7
0ed9 : fb a0 00 84 fa 8c cb 02   fc
0ee1 : c8 b1 fa 88 20 d2 ff c8   7c
0ee9 : d0 f6 20 cc ff a9 0f a2   a5
0ef1 : 08 a8 20 ba ff a9 0e a2   77
0ef9 : 52 a0 0e 20 bd ff 20 c0   01
0f01 : ff a9 0f 20 c3 ff a9 05   8a
0f09 : 20 c3 ff 20 cc ff 60 ad   b8
0f11 : c7 02 38 e9 1f 85 fa a9   82
0f19 : 7b e9 00 18 6d c6 02 85   ac
0f21 : fb 60 20 10 0f a0 00 b9   c0
0f29 : 40 03 91 fa c8 c0 3f d0   e0
0f31 : f6 60 ad ce 02 20 e5 09   67
0f39 : ad c9 02 8d 5d 0e ad c8   8c
0f41 : 02 8d 5e 0e ad cf 02 20   05
0f49 : e5 09 ad c9 02 8d 5a 0e   69
0f51 : ad c8 02 8d 5b 0e 60 ad   98
0f59 : c1 02 f0 28 c9 24 b0 24   25
0f61 : a8 ad c2 02 d9 91 0f f0   19
0f69 : 02 b0 19 8d c2 02 ad c1   32
```

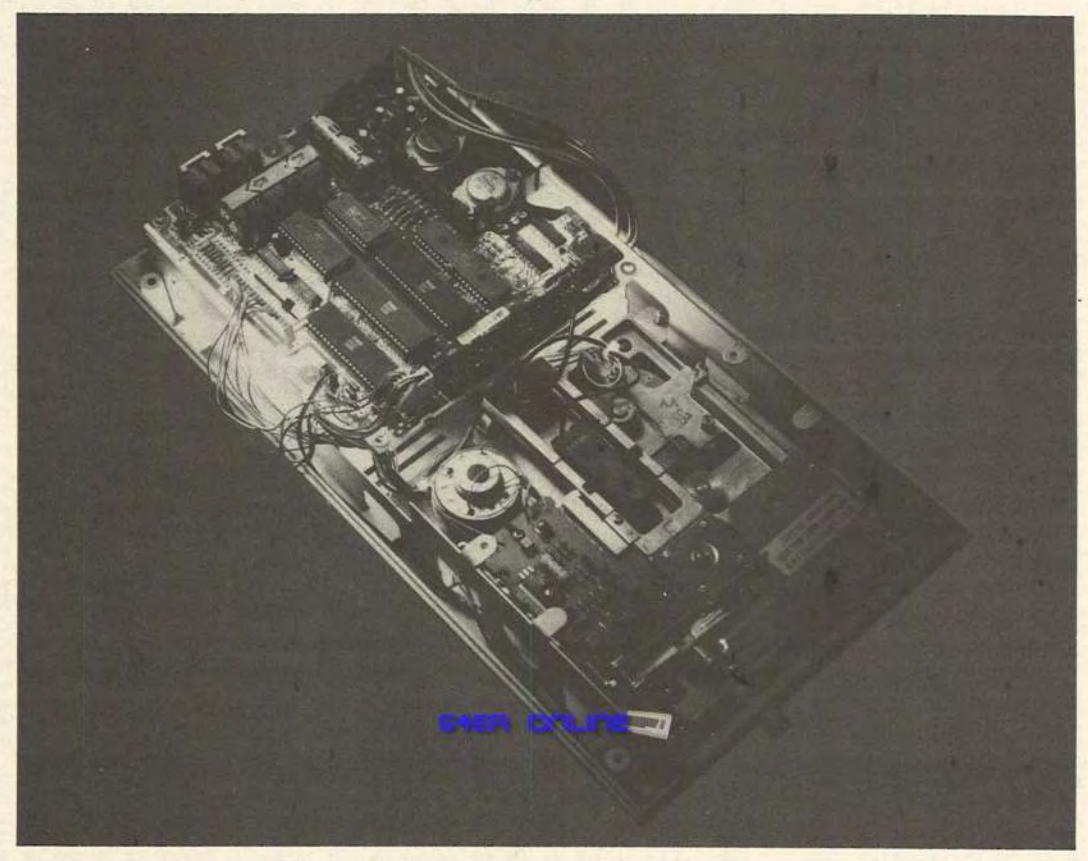
Listing 1. »Perfect View«, ein diskettenorientierter Sprite- und Zeichensatz-Editor. Bitte mit dem MSE (Seite 158) eingeben.

```
0f71 : 02 ae d0 02 9d 00 79 8d  1a
0f79 : cf 02 ad c2 02 9d 00 78  0b
0f81 : 8d ce 02 60 a9 12 8d c1  e7
0f89 : 02 a9 00 8d c2 02 4c 6f  5e
0f91 : 0f 14 14 14 14 14 14 14  8c
0f99 : 14 14 14 14 14 14 14 14  99
0fa1 : 14 14 12 12 12 12 12 12  a4
0fa9 : 12 11 11 11 11 11 11 10  a8
0fb1 : 10 10 10 10 10 a9 01 8d  3d
0fb9 : d1 02 a9 ff 8d c8 02 20  5d
0fc1 : 23 0f ad c8 02 cd c7 02  a2
0fc9 : 90 02 d0 45 ad cb 02 f0  5a
0fd1 : 06 20 33 0f 20 b4 0e ad  d1
0fd9 : c6 02 c9 03 d0 1e 20 23  38
0fe1 : 10 ee d0 02 ad 00 7e 8d  cd
0fe9 : c1 02 ad 01 7e 8d c2 02  9a
0ff1 : 20 58 0f 20 33 0f 20 60  f2
0ff9 : 0e 4c 12 10 ee c6 02 ee  bf
1001 : d0 02 ae d0 02 bd 00 78  97
1009 : 8d ce 02 bd 00 79 8d cf  d7
1011 : 02 ad c7 02 18 6d d1 02  54
1019 : 8d c7 02 20 41 0e 20 f3  fb
1021 : 0a 60 a2 7b 86 fb e8 86  6c
1029 : f9 a9 00 85 fa 85 f8 a8  b9
1031 : a2 03 b1 f8 91 fa c8 d0  96
1039 : f9 e6 fb e6 f9 ca d0 f2  a0
1041 : 60 a9 03 8d d1 02 a9 fd  b8
1049 : 8d c8 02 4c c0 0f 20 23  90
1051 : 0f ad c7 02 38 ed d4 02  b3
1059 : b0 46 ad cb 02 f0 06 20  11
1061 : 33 0f 20 b4 0e ad c6 02  28
1069 : d0 21 20 b2 10 ce d0 02  e7
1071 : ae d0 02 bd 00 78 8d c2  3f
1079 : 02 bd 00 79 8d c1 02 20  b8
1081 : 58 0f 20 33 0f 20 60 0e  5f
1089 : 4c a1 10 ce c6 02 ce d0  dd
1091 : 02 ae d0 02 bd 00 78 8d  38
1099 : ce 02 bd 00 79 8d cf 02  1f
10a1 : ad c7 02 38 ed d4 02 8d  62
10a9 : c7 02 20 41 0e 20 f3 0a  67
10b1 : 60 a2 7d 86 f9 e8 86 fb  8b
10b9 : a9 00 85 fa 85 f8 a8 a2  2b
10c1 : 03 b1 f8 91 fa c8 d0 f9  3a
10c9 : c6 fb c6 f9 ca d0 f2 60  3d
10d1 : a5 c6 f0 fc 78 20 b4 e5  dc
10d9 : 60 20 23 0f ad cb 02 f0  17
10e1 : 06 20 33 0f 20 b4 0e a9  d9
10e9 : 03 8d c6 02 a9 00 8d c7  05
10f1 : 02 20 23 10 60 20 da 10  61
10f9 : 20 41 0e 20 f3 0a a9 ce  15
1101 : 85 fc a9 02 85 fd 20 31  da
1109 : 11 a9 cf 85 fc a9 02 85  c3
1111 : fd 20 31 11 ad ce 02 8d  01
1119 : c2 02 ad cf 02 8d c1 02  d9
1121 : 20 58 0f 20 33 0f 20 60  22
1129 : 0e 20 41 0e 20 f3 0a 60  e4
1131 : a0 00 b1 fc 8d d1 02 98  7e
1139 : 91 fc 20 f3 0a 20 d1 10  d8
1141 : c9 0d d0 0a a0 00 ad d1  6b
1149 : 02 91 fc 4c 89 11 a0 00  80
1151 : 8c d1 02 4c 5e 11 20 d1  63
1159 : 10 c9 0d f0 2b c9 30 90  92
1161 : f5 c9 3a b0 f1 38 e9 30  c8
1169 : aa ad d1 02 0a 0a 18 6d  cb
1171 : d1 02 0a 8d d1 02 8a 18  ff
1179 : 6d d1 02 8d d1 02 a0 00  b1
1181 : 91 fc 20 f3 0a 4c 57 11  99
1189 : 20 f3 0a 60 20 c5 11 a9  f9
1191 : 0f a2 08 a8 20 ba ff a9  34
1199 : 00 20 bd ff 20 c0 ff a2  66
11a1 : 0f 20 c6 ff a2 16 a0 00  cf
11a9 : 18 20 f0 ff 20 cf ff 20  ce
11b1 : d2 ff a5 90 f0 f6 20 cc  df
11b9 : ff a9 0f 20 c3 ff a9 20  78
11c1 : 8d 8f 07 60 a9 20 aa e8  fc
11c9 : 9d 6f 07 ca d0 fa 60 ad  fb
11d1 : 15 d0 85 02 a9 00 8d 15  eb
11d9 : d0 20 44 e5 a9 12 20 d2  d8
11e1 : ff a2 28 bd 59 15 20 d2  57
11e9 : ff ca d0 f7 60 a5 02 8d  d7
11f1 : 15 d0 4c e3 0d 20 d0 11  35
11f9 : a9 49 8d 3a 14 a9 01 20  c4
1201 : 5d 14 a9 24 85 fb a9 fb  2e
1209 : 85 bb a9 00 85 bc a9 01  bd
1211 : 85 b7 a9 08 85 ba a9 60  73
1219 : 85 b9 20 d5 f3 a5 ba 20  d5
1221 : b4 ff a5 b9 20 96 ff a9  7f
1229 : 00 85 90 a0 03 84 fb 20  a8
1231 : a5 ff 85 fc a4 90 d0 2f  47
1239 : 20 a5 ff a4 90 d0 28 a4  3a
1241 : fb 88 d0 e9 a6 fc 20 cd  60
1249 : bd a9 20 20 d2 ff 20 a5  e0
1251 : ff a6 90 d0 12 aa f0 06  28
1259 : 20 d2 ff 4c 4f 12 a9 0d  b2
1261 : 20 d2 ff a0 02 d0 c6 20  00
1269 : 42 f6 20 d1 10 20 ee 11  48
1271 : 60 be f7 f7 f7 f7 f7 f7  3c
1279 : c7 d2 c5 d1 d2 d4 c3 f7  28
1281 : c1 de d2 c0 f7 f7 f7 f7  8d
1289 : f7 f7 f7 3e 77 77 77 77  41
1291 : 77 be c1 16 c0 3e 00 96  c6
1299 : 94 93 83 86 89 98 90 88  59
12a1 : 89 97 94 92 92 87 a0 00  55
12a9 : b9 72 12 49 57 99 c2 07  a4
12b1 : c8 c0 25 d0 f3 ae 97 12  74
12b9 : bd 98 12 8d e3 07 bd 9d  a1
12c1 : 12 8d e4 07 bd a2 12 8d  08
12c9 : e5 07 60 a9 01 8d 97 12  7e
12d1 : 20 a7 12 20 d1 10 c9 88  23
12d9 : f0 2f c9 1d d0 06 20 b6  a2
12e1 : 0f 4c d4 12 c9 9d d0 0b  71
12e9 : a9 01 8d d4 02 20 4f 10  8f
12f1 : 4c d4 12 c9 20 90 dc c9  f3
12f9 : 60 b0 d8 8d 5f 03 a9 0f  6c
1301 : 8d cb 02 20 b6 0f 4c d4  b7
1309 : 12 a9 00 8d 97 12 4c a7  2c
1311 : 12 a9 00 8d 97 12 85 c6  57
1319 : 20 a7 12 60 a9 04 8d 97  be
1321 : 12 20 a7 12 a5 c6 f0 fc  be
1329 : ad 77 02 c9 87 f0 e2 ae  34
1331 : 73 13 f0 3f c9 41 90 0a  4f
1339 : c9 47 b0 06 38 e9 37 4c  db
1341 : 4e 13 c9 30 90 2d c9 3a  9f
1349 : b0 29 38 e9 30 48 ad 5f  94
1351 : 03 0a 0a 0a 0a 8d 5f 03  ae
1359 : 68 18 6d 5f 03 8d 5f 03  35
1361 : a9 0f 8d cb 02 20 f3 0a  74
1369 : 20 23 0f a9 00 85 c6 4c  f4
1371 : 25 13 00 ae 73 13 d0 1e  45
1379 : c9 30 90 1a c9 3a b0 16  1f
1381 : a9 5f 85 fc a9 03 85 fd  9f
1389 : 20 31 11 a9 0f 8d cb 02  4c
1391 : 20 b6 0f 4c 25 13 20 d1  69
1399 : 10 c9 1d d0 06 20 b6 0f  4a
13a1 : 4c 25 13 c9 9d d0 08 a9  52
13a9 : 01 8d d4 02 20 4f 10 4c  3c
13b1 : 25 13 48 20 c5 11 a2 16  12
13b9 : a0 00 20 0c e5 68 20 d2  ab
13c1 : ff a9 3e 20 d2 ff a2 00  e0
13c9 : 86 ff e0 1e f0 05 a9 af  88
13d1 : 20 d2 ff 20 d1 10 a6 ff  96
13d9 : c9 0d f0 4c c9 14 d0 22  b3
13e1 : e0 00 f0 ef e0 1e f0 05  c8
13e9 : a9 9d 20 d2 ff a9 [illegible] 20  c7
13f1 : d2 ff a9 20 a6 ff 9d 72  f7
13f9 : 07 ca 9d 3a 14 86 ff 4c  22
1401 : cb 13 e0 1e f0 cd c9 20  37
1409 : 90 c9 c9 22 f0 c5 c9 60  5a
1411 : b0 c1 a8 a9 9d 20 d2 ff  27
1419 : 98 a6 ff 9d 3a 14 20 d2  22
1421 : ff e6 ff a6 ff 4c cb 13  20
1429 : 60 a9 43 20 b3 13 a6 ff  a1
1431 : e0 00 f0 04 8a 20 5d 14  15
1439 : 60 54 2e 46 2e 2c 48 4f  1c
1441 : 45 4c 4c 45 20 31 30 2c  0d
1449 : 41 2d 32 38 30 33 20 53  78
1451 : 43 48 57 41 52 5a 45 4e  60
1459 : 42 41 43 48 85 ff a9 0f  33
1461 : a2 08 a8 20 ba ff a9 00  88
1469 : 20 bd ff 20 c0 ff a2 0f  20
1471 : 20 c9 ff a2 00 8a 48 bd  bb
1479 : 3a 14 20 d2 ff 68 aa e8  df
1481 : e4 ff d0 f1 20 cc ff a9  93
1489 : 0f 20 c3 ff 60 2d 30 20  0a
1491 : 8d 2d 31 20 8d 2d 32 20  51
1499 : 8d 2d 33 20 8d 2d 34 20  e1
14a1 : 8d 2d 35 20 8d 2d 36 20  72
14a9 : 8d 2d 37 20 8d 2d 38 20  02
14b1 : 8d 2d 39 20 8d 2d 30 31  8d
14b9 : 8d 2d 31 31 8d 2d 32 31  bd
14c1 : 8d 2d 33 31 8d 2d 34 31  4d
14c9 : 8d 2d 35 31 8d 2d 36 31  de
14d1 : 8d 2d 37 31 8d 2d 38 31  6e
14d9 : 8d 2d 39 31 8d 2d 30 32  d9
14e1 : 20 35 34 33 32 31 30 39  ef
14e9 : 38 37 36 35 34 33 32 31  f9
14f1 : 30 39 38 37 36 35 34 33  f7
14f9 : 32 31 30 39 38 37 36 35  78
1501 : 34 33 32 31 20 20 20 20  45
1509 : 20 33 33 33 33 33 33 32  f4
1511 : 32 32 32 32 32 32 32 32  11
1519 : 32 31 31 31 31 31 31 31  1a
1521 : 31 31 31 20 20 20 20 20  ff
1529 : 20 20 20 20 20 20 20 20  29
1531 : 92 20 20 20 20 20 20 20  a3
1539 : 20 20 20 20 20 20 20 20  39
1541 : 20 20 20 20 20 20 20 20  41
1549 : 20 20 20 20 20 20 20 20  49
1551 : 3a 45 4d 41 4e 4b 53 49  c8
1559 : 44 3c 3c 20 52 45 47 4e  d8
1561 : 49 52 48 45 55 46 20 53  3d
1569 : 41 4d 4f 48 54 20 4e 4f  4c
1571 : 56 20 57 45 49 56 20 54  c6
1579 : 43 45 46 52 45 50 20 3e  0f
1581 : 3e 12 01 00 a9 00 8d ce  77
1589 : 02 8d 84 15 a9 12 8d cf  17
1591 : 02 ad c6 02 85 ff a9 04  63
1599 : 8d c6 02 20 33 0f 20 60  fb
15a1 : 0e a5 ff 8d c6 02 ae d0  0c
15a9 : 02 bd 00 78 8d ce 02 bd  6c
15b1 : 00 79 8d cf 02 20 d0 11  52
15b9 : 20 44 e5 a2 f4 bd 8e 14  69
15c1 : 20 d2 ff ca d0 f7 a2 01  fd
15c9 : a0 09 18 20 f0 ff a9 12  d2
15d1 : 20 d2 ff a2 00 bd 90 7f  de
15d9 : 20 d2 ff e8 e0 17 d0 f5  75
15e1 : a9 0d 20 d2 ff a9 01 8d  e0
15e9 : 83 15 a2 04 a9 07 85 fb  01
15f1 : a9 c3 18 6d 83 15 85 fa  1d
15f9 : ad 84 15 29 07 d0 01 e8  20
1601 : a8 b9 4f 16 a0 00 3d 00  1b
1609 : 7f f0 05 a9 57 4c 13 16  c7
1611 : a9 a0 91 fa a5 fa 38 e9  b5
1619 : 28 85 fa a5 fb e9 00 85  91
1621 : fb ee 84 15 ad 83 15 a8  f4
1629 : 88 b9 92 0f 18 69 01 cd  81
1631 : 84 15 d0 c4 a9 00 8d 84  e6
1639 : 15 e8 ee 83 15 ad 83 15  e5
1641 : c9 24 d0 a8 20 d1 10 20  77
1649 : ee 11 20 57 16 60 01 02  1f
1651 : 04 08 10 20 40 80 a0 00  ec
1659 : 98 99 00 7a 99 00 7f c8  36
1661 : d0 f7 60 00 00 00 00 18  75
1669 : 00 00 18 00 00 ff 00 01  71
1671 : ff 80 02 18 40 06 18 60  89
1679 : 06 00 60 06 00 60 1f 81  db
1681 : f8 1f 81 f8 06 00 60 06  76
1689 : 00 60 06 18 60 02 18 40  35
1691 : 01 ff 80 00 ff 00 00 18  e2
1699 : 00 00 18 00 00 00 00 00  a0
16a1 : 00 00 00 00 00 a9 fb 2d  39
16a9 : 17 d0 8d 17 d0 a9 03 8d  f0
16b1 : 97 12 20 a7 12 a9 2d 8d  8d
16b9 : 05 d0 a9 d9 8d 04 d0 a9  5b
16c1 : 00 8d a4 16 8d a5 16 8d  ed
16c9 : 10 d0 a9 07 8d 15 d0 a9  a5
16d1 : fb 2d 1d d0 8d 1d d0 20  09
16d9 : f3 0a a5 c6 d0 6f 20 ef  fc
16e1 : 17 ad 00 dc aa 29 01 d0  04
16e9 : 0b ac a4 16 f0 37 20 d1  23
16f1 : 17 4c 26 17 8a 29 02 d0  36
16f9 : 0d ac a4 16 c0 07 f0 25  9b
1701 : 20 e0 17 4c 26 17 8a 29  78
1709 : 04 d0 0b ac a5 16 f0 15  c7
1711 : 20 d8 18 4c 26 17 8a 29  c5
1719 : 08 d0 0a ac a5 16 c0 07  be
1721 : f0 03 20 f1 18 ad 00 dc  82
1729 : 29 10 d0 ab 20 6d 08 20  d2
1731 : 6d 08 20 6d 08 ae a5 16  11
1739 : bd 26 09 ac a4 16 59 5c  fa
1741 : 03 99 5c 03 20 6d 08 20  56
1749 : 6d 08 4c d8 16 20 d1 10  b2
1751 : c9 86 d0 03 4c fd 17 c9  96
1759 : 49 d0 06 20 0a 19 4c d8  dc
1761 : 16 c9 93 d0 06 20 18 19  4f
1769 : 4c d8 16 c9 9d d0 06 20  99
1771 : 2b 19 4c d8 16 c9 1d d0  1d
1779 : 06 20 44 19 4c d8 16 c9  3b
1781 : 03 d0 06 20 41 0e 4c d8  d9
1789 : 16 c9 91 d0 06 20 5d 19  0b
1791 : 4c d8 16 c9 11 d0 06 20  f8
1799 : 73 19 4c d8 16 c9 2c d0  c9
17a1 : 06 20 c3 19 4c d8 16 c9  43
17a9 : 2e d0 06 20 87 19 4c d8  e9
17b1 : 16 aa 20 d8 0b a5 ff d0  bf
17b9 : 14 e0 58 d0 06 20 e8 19  a5
17c1 : 4c d8 16 e0 59 d0 06 20  8f
17c9 : fd 19 4c d8 16 4c d8 16  d4
17d1 : ce a4 16 a2 08 ce 05 d0  78
17d9 : 20 ef 17 ca d0 f7 60 ee  3c
17e1 : a4 16 a2 08 ee 05 d0 20  d5
17e9 : ef 17 ca d0 f7 60 8a 48  6e
17f1 : 20 61 09 20 61 09 20 61  aa
17f9 : 09 68 aa 60 a9 00 8d 97  ed
1801 : 12 20 a7 12 a9 03 8d 15  63
1809 : d0 20 23 0f 20 57 16 ad  05
1811 : c6 02 f0 3a ad c7 02 c9  10
1819 : 03 b0 33 ae d0 02 ca bd  db
1821 : 00 78 8d ce 02 bd 00 79  9b
1829 : 8d cf 02 ce c6 02 20 33  5c
1831 : 0f 20 b4 0e ee c6 02 ae  ca
1839 : d0 02 bd 00 78 8d ce 02  ad
1841 : bd 00 79 8d cf 02 a9 0f  e0
1849 : 8d cb 02 4c 41 0e ad c6  8f
1851 : 02 c9 03 f0 f3 ad c7 02  e6
1859 : c9 fb 90 ea ad d0 02 e8  dc
1861 : bd 00 78 8d ce 02 bd 00  e2
1869 : 79 8d cf 02 ee c6 02 20  4a
```

```
1871 : 33 0f 20 b4 0e ce c6 02   41
1879 : 4c 38 18 20 10 0f a0 00   e7
1881 : a2 00 a5 fa f0 07 c9 01   5c
1889 : f0 03 4c 92 18 a9 02 85   42
1891 : fa b1 fa 9d 40 03 a5 fa   7f
1899 : 18 69 01 85 fa a5 fb 69   f6
18a1 : 00 85 fb e8 e0 3f d0 da   81
18a9 : 60 20 10 0f a0 00 a2 00   94
18b1 : a5 fa f0 07 c9 01 f0 03   5f
18b9 : 4c c0 18 a9 02 85 fa bd   54
18c1 : 40 03 91 fa a5 fa 18 69   ac
18c9 : 01 85 fa a5 fb 69 00 85   16
18d1 : fb e8 e0 3f d0 da 60 ce   63
18d9 : a5 16 a2 08 ad 04 d0 d0   13
18e1 : 05 a9 00 8d 10 d0 ce 04   37
18e9 : d0 20 ef 17 ca d0 ed 60   54
18f1 : ee a5 16 a2 08 ee 04 d0   35
18f9 : ad 04 d0 d0 05 a9 04 8d   bf
1901 : 10 d0 20 ef 17 ca d0 ed   66
1909 : 60 a2 08 a9 ff 5d 5b 03   50
1911 : 9d 5b 03 ca d0 f5 60 a2   f9
1919 : 08 a9 00 9d 5b 03 ca d0   44
1921 : f8 60 47 41 4c 4c 59 4d   6a
1929 : 45 44 a2 08 a9 80 3d 5b   84
1931 : 03 18 2a 2a 85 ff bd 5b   16
1939 : 03 0a 05 ff 9d 5b 03 ca   d9
1941 : d0 ea 60 a2 08 a9 01 3d   3f
1949 : 5b 03 18 6a 6a 85 ff bd   c7
1951 : 5b 03 4a 05 ff 9d 5b 03   c1
1959 : ca d0 ea 60 ad 5c 03 48   ac
1961 : a2 00 bd 5d 03 9d 5c 03   b3
1969 : e8 e0 07 d0 f5 68 8d 63   3d
1971 : 03 60 ad 63 03 48 a2 07   87
1979 : bd 5b 03 9d 5c 03 ca d0   03
1981 : f7 68 8d 5c 03 60 20 b8   c0
1989 : 19 a0 00 a2 00 bd 5c 03   ac
1991 : 39 26 09 f0 03 bd 4f 16   c5
1999 : 19 23 19 99 23 19 e8 e0   1e
19a1 : 08 d0 ea c8 c0 08 d0 e3   3c
19a9 : 4c ac 19 a0 08 b9 22 19   af
19b1 : 99 5b 03 88 d0 f7 60 a0   59
19b9 : 08 a9 00 99 22 19 88 d0   78
19c1 : f8 60 20 b8 19 a0 00 a2   e4
19c9 : 00 bd 5c 03 39 4f 16 f0   68
19d1 : 03 bd 26 09 19 23 19 99   a0
19d9 : 23 19 e8 e0 08 d0 ea c8   23
19e1 : c0 08 d0 e3 4c ac 19 20   25
19e9 : b8 19 a2 00 a0 07 bd 5c   c8
19f1 : 03 99 23 19 e8 88 10 f6   ae
19f9 : 20 ac 19 60 a2 08 bd 5b   da
1a01 : 03 20 0c 1a 9d 5b 03 ca   b1
1a09 : d0 f4 60 48 a0 00 84 ff   91
1a11 : 68 48 39 26 09 f0 07 b9   58
1a19 : 4f 16 05 ff 85 ff c8 c0   b2
1a21 : 08 d0 ed 68 a5 ff 60 a9   49
1a29 : 02 8d 97 12 20 a7 12 20   e2
1a31 : 7c 18 a9 2d 8d 05 d0 a9   61
1a39 : 11 8d 04 d0 a9 fb 2d 17   89
1a41 : d0 8d 17 d0 a9 07 8d 15   eb
1a49 : d0 a9 00 8d 10 d0 ad c5   69
1a51 : 02 c9 08 d0 08 a9 fb 2d   6c
1a59 : 1d d0 4c 68 1a a9 09 8d   2d
1a61 : 04 d0 a9 04 0d 1d d0 8d   d0
1a69 : 1d d0 a9 00 8d a4 16 8d   ca
1a71 : a5 16 20 f3 0a 20 f3 0a   2d
1a79 : a5 c6 f0 03 4c 00 1b ad   aa
1a81 : 00 dc 48 29 04 d0 12 ad   91
1a89 : a5 16 f0 0d 20 d8 18 ad   9c
1a91 : c5 02 c9 08 f0 03 20 d8   24
1a99 : 18 68 48 29 08 d0 1e ad   f7
1aa1 : a5 16 ae c5 02 e0 08 f0   df
1aa9 : 03 18 69 01 c9 17 b0 2b   a1
1ab1 : 20 f1 18 ad c5 02 c9 08   29
1ab9 : f0 03 20 f1 18 68 48 29   a9
1ac1 : 01 d0 08 ad a4 16 f0 03   a7
1ac9 : 20 d1 17 68 48 29 02 d0   1c
1ad1 : 0a ad a4 16 c9 14 f0 03   a5
1ad9 : 20 e0 17 68 29 10 d0 95   be
1ae1 : 20 e6 1b aa ad a5 16 29   43
1ae9 : 07 a8 b9 26 09 5d 40 03   fa
1af1 : 9d 40 03 20 a3 20 f3 0a   6b
1af9 : 0a 20 f3 0a 4c 76 1a 20   73
1b01 : d1 10 c9 85 d0 03 4c 00   54
1b09 : 1c c9 31 f0 0f c9 32 f0   5e
1b11 : 0b c9 33 f0 07 c9 34 f0   5d
1b19 : 03 4c 20 1b 4c 82 1c c9   8b
1b21 : 03 d0 06 20 7c 18 4c 76   b9
1b29 : 1a aa 20 d8 0b a5 ff f0   7b
1b31 : 03 4c 76 1a 8a c9 41 d0   d9
1b39 : 06 20 b5 1c 4c 76 1a c9   b5
1b41 : 24 d0 06 20 f6 11 4c 76   69
1b49 : 1a c9 45 d0 09 20 8d 11   9d
1b51 : 20 d1 10 4c 76 1a c9 43   cd
1b59 : d0 06 20 2a 14 4c 76 1a   2b
1b61 : c9 5e d0 0b a9 03 4d 17   05
1b69 : d0 8d 17 d0 4c 76 1a c9   54
1b71 : 5f d0 0b a9 03 4d 1d d0   e1
1b79 : 8d 1d d0 4c 76 1a c9 49   44
1b81 : d0 06 20 23 1e 4c 76 1a   13
1b89 : c9 93 d0 06 20 18 1e 4c   e5
1b91 : 76 1a c9 58 d0 06 20 c1   d3
1b99 : 1d 4c 76 1a c9 59 d0 06   74
1ba1 : 20 e5 1d 4c 76 1a c9 1d   1e
1ba9 : d0 06 20 89 1d 4c 76 1a   f8
1bb1 : c9 9d d0 06 20 51 1d 4c   d7
1bb9 : 76 1a c9 91 d0 06 20 fb   97
1bc1 : 1c 4c 76 1a c9 11 d0 06   59
1bc9 : 20 21 1d 4c 76 1a c9 4c   43
1bd1 : d0 06 20 71 1e 4c 76 1a   2d
1bd9 : c9 53 d0 06 20 99 1e 4c   21
1be1 : 76 1a 4c 76 1a a9 00 18   65
1be9 : 6d a4 16 6d a4 16 6d a4   d6
1bf1 : 16 85 ff ad a5 16 4a 4a   48
1bf9 : 4a 18 65 ff 85 ff 60 a9   d6
1c01 : 00 8d 97 12 20 a7 12 a9   cb
1c09 : 03 8d 15 d0 20 aa 18 20   2a
1c11 : 57 16 20 41 0e ad c6 02   11
1c19 : f0 3a ad c7 02 c9 20 b0   db
1c21 : 33 ae d0 02 ca bd 00 78   ab
1c29 : 8d ce 02 bd 00 79 8d cf   f7
1c31 : 02 ce c6 02 20 33 0f 20   a4
1c39 : b4 0e ee c6 02 ae d0 02   66
1c41 : bd 00 78 8d ce 02 bd 00   c2
1c49 : 79 8d cf 02 a9 0f 8d cb   9e
1c51 : 02 4c f3 0a ad c6 02 c9   64
1c59 : 03 f0 f1 ad c7 02 c9 df   7a
1c61 : 90 ea ae d0 02 e8 bd 00   8a
1c69 : 78 8d ce 02 bd 00 79 8d   79
1c71 : cf 02 ee c6 02 20 33 0f   e2
1c79 : 20 b4 0e ce c6 02 4c 3e   7b
1c81 : 1c 38 e9 31 48 20 e6 1b   b1
1c89 : 68 aa ad a5 16 29 07 4a   c2
1c91 : a8 b9 ad 1c 3d b1 1c a6   24
1c99 : ff 85 ff b9 ad 1c 49 ff   73
1ca1 : 3d 40 03 18 65 ff 9d 40   0f
1ca9 : 03 4c 76 1a c0 30 0c 03   77
1cb1 : 00 aa ff 55 ad c5 02 c9   55
1cb9 : 10 d0 14 ad 04 d0 18 69   e6
1cc1 : 08 8d 04 d0 a9 fb 2d 1d   14
1cc9 : d0 8d 1d d0 4c ea 1c ad   a9
1cd1 : a5 16 6a 90 03 20 d8 18   f3
1cd9 : ad 04 d0 38 e9 08 8d 04   e1
1ce1 : d0 a9 04 0d 1d d0 8d 1d   f1
1ce9 : d0 a9 18 4d c5 02 8d c5   6c
1cf1 : 02 20 2e 09 4c f3 0a 47   cb
1cf9 : 41 4c ad 40 03 48 ad 41   7f
1d01 : 03 48 ad 42 03 48 a2 00   d9
1d09 : bd 43 03 9d 40 03 e8 e0   5e
1d11 : 3c d0 f5 68 8d 7e 03 68   e9
1d19 : 8d 7d 03 68 8d 7c 03 60   bc
1d21 : ad 7c 03 8d f8 1c ad 7d   a1
1d29 : 03 8d f9 1c ad 7e 03 8d   eb
1d31 : fa 1c a2 3d bd 3f 03 9d   a7
1d39 : 42 03 ca d0 f7 ad fa 1c   da
1d41 : 8d 42 03 ad f9 1c 8d 41   9f
1d49 : 03 ad f8 1c 8d 40 03 60   8c
1d51 : a9 00 8d c1 02 ae c1 02   36
1d59 : bd 40 03 29 80 18 2a 2a   e2
1d61 : 85 ff bd 42 03 18 2a 05   41
1d69 : ff 9d 42 03 bd 41 03 2a   6e
1d71 : 9d 41 03 bd 40 03 2a 9d   27
1d79 : 40 03 ad c1 02 18 69 03   6b
1d81 : 8d c1 02 c9 3f d0 ce 60   1f
1d89 : a9 00 8d c1 02 ae c1 02   6e
1d91 : bd 42 03 29 01 18 6a 6a   a5
1d99 : 85 ff bd 40 03 18 6a 05   3a
1da1 : ff 9d 40 03 bd 41 03 6a   a6
1da9 : 9d 41 03 bd 42 03 6a 9d   80
1db1 : 42 03 ad c1 02 18 69 03   a5
1db9 : 8d c1 02 c9 3f d0 ce 60   57
1dc1 : a2 00 a0 3c bd 40 03 99   30
1dc9 : c0 03 bd 41 03 99 c1 03   ac
1dd1 : bd 42 03 99 c2 03 e8 e8   5d
1dd9 : e8 88 88 88 e0 3f d0 e4   4e
1de1 : 20 0c 1e 60 a2 00 bd 42   41
1de9 : 03 48 bd 40 03 20 0c 1a   1d
1df1 : 9d 42 03 bd 41 03 20 0c   ed
1df9 : 1a 9d 41 03 68 20 0c 1a   7f
1e01 : 9d 40 03 e8 e8 e8 e0 3f   74
1e09 : d0 dc 60 a2 3f bd bf 03   9b
1e11 : 9d 3f 03 ca d0 f7 60 a2   fb
1e19 : 3f a9 00 9d 3f 03 ca d0   b9
1e21 : fa 60 a2 3f bd 3f 03 49   50
1e29 : ff 9d 3f 03 ca d0 f5 60   f3
1e31 : 85 02 20 b3 13 a5 ff f0   76
1e39 : 36 a9 05 a8 a2 08 20 ba   fb
1e41 : ff a5 ff c9 10 90 02 a9   2d
1e49 : 10 aa a9 2c 9d 3a 14 e8   6c
1e51 : a9 50 9d 3a 14 e8 a9 2c   58
1e59 : 9d 3a 14 e8 a5 02 9d 3a   8b
1e61 : 14 e8 8a a2 3a a0 14 20   19
1e69 : bd ff 20 c0 ff a9 05 60   68
1e71 : a9 52 20 31 1e f0 f8 a2   04
1e79 : 05 20 c6 ff a2 00 20 cf   8a
1e81 : ff 20 cf ff 20 cf ff 9d   40
1e89 : 40 03 e8 e0 3f d0 f5 20   34
1e91 : cc ff a9 05 20 c3 ff 60   49
1e99 : a9 57 20 31 1e f0 f8 a2   af
1ea1 : 05 20 c9 ff a2 00 a9 40   7a
1ea9 : 20 d2 ff a9 03 20 d2 ff   e4
1eb1 : bd 40 03 20 d2 ff e8 e0   e6
1eb9 : 3f d0 f5 4c 90 1e a9 57   b7
1ec1 : 20 31 1e f0 d2 a2 05 20   b6
1ec9 : c9 ff a2 00 bd 7a 1f 20   a7
1ed1 : d2 ff e8 e0 28 d0 f5 a2   1f
1ed9 : 00 bd 00 78 20 d2 ff bd   db
1ee1 : 00 79 20 d2 ff bd 00 7b   e5
1ee9 : 20 d2 ff bd 00 7c 20 d2   34
1ef1 : ff bd 00 7d 20 d2 ff bd   93
1ef9 : 00 7e 20 d2 ff e8 d0 d9   d9
1f01 : a2 00 bd bc 02 20 d2 ff   17
1f09 : e8 e0 17 d0 f5 ad 20 d0   30
1f11 : 20 d2 ff ad 21 d0 20 d2   0f
1f19 : ff 4c 90 1e 60 a9 52 20   03
1f21 : 31 1e f0 f8 a2 05 20 c6   1d
1f29 : ff a2 28 20 cf ff ca d0   51
1f31 : fa a2 00 20 cf ff 9d 00   f4
1f39 : 78 20 cf ff 9d 00 79 20   b5
1f41 : cf ff 9d 00 7b 20 cf ff   6f
1f49 : 9d 00 7c 20 cf ff 9d 00   7d
1f51 : 7d 20 cf ff 9d 00 7e e8   78
1f59 : d0 d9 a2 00 20 cf ff 9d   7a
1f61 : bc 02 e8 e0 17 d0 f5 20   85
1f69 : cf ff 8d 20 d0 20 cf ff   ec
1f71 : 8d 21 d0 20 90 1e 4c e3   ba
1f79 : 0d 01 08 24 08 c3 07 50   e9
1f81 : 45 52 46 45 43 54 20 56   2e
1f89 : 49 45 57 20 56 4f 4e 20   a8
1f91 : 54 48 2e 20 46 55 45 48   4e
1f99 : 52 49 4e 47 45 52 00 00   f3
1fa1 : 00 58 20 a0 e5 78 a9 31   15
1fa9 : 8d 14 03 a9 ea 8d 15 03   ac
1fb1 : 58 a2 09 bd f1 1f 9d ff   e3
1fb9 : 7f ca d0 f7 a9 00 8d 20   e1
1fc1 : d0 8d 21 d0 20 44 e5 a9   c9
1fc9 : fb 85 fa a9 1f 85 fb a0   ca
1fd1 : 00 b1 fa f0 13 20 d2 ff   04
1fd9 : a5 fa 18 69 01 85 fa a5   a2
1fe1 : fb 69 00 85 fb 4c d0 1f   e5
1fe9 : a5 c6 f0 fc c6 c6 4c 0d   bb
1ff1 : 0b a2 1f a2 1f c3 c2 cd   20
1ff9 : 38 30 11 11 11 11 11 12   b2
2001 : 1c 20 20 20 20 20 20 20   fd
2009 : 20 20 20 20 20 20 20 20   09
2011 : 20 20 20 20 20 20 20 20   11
2019 : 20 20 20 20 20 20 20 20   19
2021 : 20 20 20 20 20 20 20 20   21
2029 : 20 05 20 20 20 20 20 20   9c
2031 : 20 20 20 20 20 20 20 20   31
2039 : 20 20 20 20 20 20 20 20   39
2041 : 20 20 20 20 20 20 20 20   41
2049 : 20 20 20 20 20 20 20 20   49
2051 : 20 20 1c 20 20 20 20 20   50
2059 : 20 20 20 20 20 20 20 20   59
2061 : 20 20 20 20 20 20 20 20   61
2069 : 20 20 20 20 20 20 20 20   69
2071 : 20 20 20 20 20 20 20 20   71
2079 : 20 20 20 92 05 11 11 20   61
2081 : 20 20 20 20 20 20 20 50   e2
2089 : 20 45 20 52 20 46 20 45   dd
2091 : 20 43 20 54 20 20 20 56   16
2099 : 20 49 20 45 20 57 0d 0d   1a
20a1 : 20 20 20 20 20 20 20 20   a1
20a9 : 20 20 20 20 20 20 20 20   a9
20b1 : 20 20 56 4f 4e 0d 0d 20   23
20b9 : 20 20 20 20 20 20 20 20   b9
20c1 : 20 20 54 48 4f 4d 41 53   1b
20c9 : 20 46 55 45 48 52 49 4e   e3
20d1 : 47 45 52 0d 0d 0d 12 1c   ab
20d9 : 20 20 20 20 20 20 20 20   d9
20e1 : 20 20 20 20 20 20 20 20   e1
20e9 : 20 20 20 20 20 20 20 20   e9
20f1 : 20 20 20 20 20 20 20 20   f1
20f9 : 20 20 20 20 20 20 20 20   f9
2101 : 05 20 20 20 20 20 20 20   e6
2109 : 20 20 20 20 20 20 20 20   09
2111 : 20 20 20 20 20 20 20 20   11
2119 : 20 20 20 20 20 20 20 20   19
2121 : 20 20 20 20 20 20 20 20   21
2129 : 20 1c 20 20 20 20 20 20   27
2131 : 20 20 20 20 20 20 20 20   31
2139 : 20 20 20 20 20 20 20 20   39
2141 : 20 20 20 20 20 20 20 20   41
2149 : 20 20 20 20 20 20 20 20   49
2151 : 20 20 00 00 00 00 00 00   82
```

Listing 1. »Perfect View« (Schluß)

In die Geheimnisse der Floppy eingetaucht



Die Diskettenlaufwerke 1541, 1570 und 1571 sind Renner unter den Massenspeichern. Doch mit der passenden Literatur hapert es. Deshalb beschränken sich die meisten Anwender auf das Speichern und Laden von Programmen. Mit diesem Kurs lernen Sie, Ihr Laufwerk effektiver einzusetzen und es sogar zu manipulieren.

Daß die 1541 (die Angaben gelten auch für die 1570/1571 im 1541-Modus) ein sehr wandelbares Gerät ist, werden die meisten Benutzer wohl wissen oder zumindest erahnen. Man denke ja nur an den »Kleinkrieg« zwischen Softwareherstellern und Softwarepiraten, die sich gegenseitig das Leben schwer machen. Die meisten »Schlachten« liefert man sich hier im Inneren der Floppy-Station, die viel raffiniertere Methoden des Programmschutzes anbietet als der Commodore 64 selbst.

Doch wie bei so vielen Dingen in der Commodore-Welt sind auch hier die Informationen rar, beziehungsweise in den Handbüchern gar nicht vorhanden. So wollen wir uns mit Ihnen an das Floppy-Laufwerk heran- und vorsichtig hineintasten. Angefangen bei grundlegenden Informationen über den Diskettenaufbau und den Befehlssatz des Laufwerks werden wir Schritt für Schritt in dessen Möglichkeiten zur Programmierung und Manipulation hinabtauchen. Was wird benötigt?

Nun, außer einem C 64 (C 128) und einer 1541/70/71, »nur« Basic-Erfahrungen, grundlegende Kenntnisse in Maschinensprache und ein wenig Geduld.

Bevor wir jedoch mit unserer ersten Tauchfahrt beginnen, tippen Sie bitte das beigefügte Programm EDDI (Listing 1) ein, sofern Sie nicht über einen eigenen Disk-Monitor verfügen. Auf die Bedienung von EDDI wird im einzelnen noch eingegangen. Natürlich können Sie alle hier angeführten Beispiele auch mit dem wesentlich leistungsfähigeren und komfortableren Monitor »Disc-Scanner 40« (Seite 40) durchführen.

Sehen wir uns jetzt erst einmal eine Diskette an. Die folgenden Erläuterungen beziehen sich auf eine formatierte Diskette.

Aufbau einer Diskette

Eine Diskette ist in 35 konzentrische Spuren (englisch: Tracks) aufgeteilt. Jede dieser Spuren enthält wiederum eine bestimmte Anzahl von Sektoren, die von außen nach innen abnimmt. Die genauen Zahlenverhältnisse stehen in Tabelle 1.

Die Spuren sind, beginnend mit der äußeren Spur, von 1 bis 35 durchnumeriert. Die Sektoren sind auf den Spuren in numerischer Reihenfolge gegen den Uhrzeigersinn angeordnet. Jeder Sektor enthält einen Block, das sind 256 Byte, an Information. Es kann jeder der 683 Blöcke auf der Diskette durch Angabe der jeweiligen Spur- und Sektornummer aufgerufen werden. Allerdings stehen davon dem Benutzer normalerweise nur 664 (1328) Blocks zur Verfügung, da das Betriebssystem der Floppystation die Spur 18 (18 und 53) für sich beschlagnahmt (für die 1571 gilt natürlich jeweils die doppelte Kapazität im C 128-Modus).

Für die nun folgenden Versuche wäre es sinnvoll, eine Diskette neu zu formatieren, mit der wir ein bißchen »spielen« können. Sehen wir uns nun erst einmal das Directory an (LOAD "$",8):

In der ersten Zeile stehen die Drive-Nummer (hier immer 0) und der Name der Diskette sowie die ID und das Formatkennzeichen (genaueres später).

Die zweite Zeile enthält, da sich kein File auf der Diskette befindet, die Meldung »664 BLOCKS FREE«.

Spur 01 bis 17	21 Sektoren
Spur 18 bis 24	19 Sektoren
Spur 25 bis 30	18 Sektoren
Spur 31 bis 35	17 Sektoren
Spur 36 bis 52	21 Sektoren (nur 1571)
Spur 53 bis 59	19 Sektoren (nur 1571)
Spur 60 bis 65	18 Sektoren (nur 1571)
Spur 66 bis 70	17 Sektoren (nur 1571)

Tabelle 1. Spuren und Sektoren des 1541-Diskettenformates

Da sich diese Informationen auf der schon erwähnten Spur 18 befinden, wollen wir uns diese Spur mit EDDI gleich einmal etwas genauer ansehen. Laden Sie den Editor und legen Sie unsere »Spieldiskette« ein; danach starten Sie mit RUN.

Als Kommando tippen Sie <F3> für »BLOCK LESEN«.

Danach geben Sie, durch Komma getrennt, die Spur und Sektornummer des gewünschten Blocks ein; in unserem Fall »18,0«.

Nach dem Ladevorgang meldet sich EDDI mit Byte 0 der ersten von 16 Seiten, zu je 16 Byte. Drücken Sie jetzt <RETURN>, um die erste Seite anzuzeigen, welche wir nun betrachten wollen.

Es sollte vielleicht erwähnt werden, daß die Zählung von Blöcken und Bytes grundsätzlich bei Null beginnt. Den geladenen Block bezeichnet man als BAM (Block Availability Map), auf deutsch etwa »Blockbelegungsplan«. Dieser Plan gibt an, welche Blöcke auf der Diskette frei und welche schon beschrieben sind. Ferner enthält er den Namen der Diskette, die ID, das Formatkennzeichen und den Beginn des Directory.

Die ersten beiden Byte (0,1) dieses Blocks enthalten Spur und Sektor des ersten Directory-Blocks; normalerweise »18,1« (siehe auch Tabelle 2). Byte 2 enthält das Formatkennzeichen (hier 65, beziehungsweise »A«). Zur Erklärung: Commodore stellt bzw. stellte verschiedene Laufwerke her, zum Beispiel die 1541, 4040, 8050 und 8250. Diese Laufwerke unterscheiden sich fast alle im Aufzeichnungsformat, das heißt Anzahl und Verteilung der Spuren und Sektoren; so hat die CBM 8050 77 Spuren mit bis zu 29 Sektoren, was deren höhere Speicherkapazität zur Folge hat. Solche Disketten können verständlicherweise von der 1541 weder gelesen noch beschrieben werden. Am Formatkennzeichen »A« erkennt die 1541 nun Disketten ihres eigenen Formats; ist dieses nicht identisch, so beschwert sich die Floppy-Station mit einer Fehlermeldung. Eine Ausnahme dieser Regel bildet die Lesekompatibilität, die besagt, daß eine »fremde« Diskette zwar gelesen, aber nicht beschrieben werden kann (zum Beispiel eine Diskette des 4040- auf dem 1541-Laufwerk).

Byte 3 steht generell auf Null, da es bei der 1541 keine Funktion erfüllt.

Erste Versuche mit EDDI, dem Disk-Monitor/Editor

Die Byte 4 bis 143 enthalten nun die eigentliche BAM, deren Format ein wenig kompliziert ist: Für jede Spur sind 4 Byte reserviert, wobei das jeweils erste Byte die Anzahl der noch freien Blöcke auf dieser Spur angibt. Die folgenden 3 Byte müssen wir als eine Gesamtheit von 24 Bit betrachten, wobei jedes gesetzte Bit einen freien Block signalisiert; siehe auch Tabelle 3.

Byte	Bedeutung
000	enthält 18 ($12); Spurnummer für Directory
001	enthält 1 ($01); Startsektor für Directory
002	enthält 65 ($41); Formatkennzeichen »A«
003	Flag für doppelseitige Disketten (1 = doppelseitige Disk, keine Bedeutung im 1541-Modus)
004	Anzahl der freien Blöcke/Sektoren für Spur 1
005–007	Bitmuster der Blockbelegung für Spur 1: Bit = 1 bedeutet »Sektor/Block frei« Bit = 0 bedeutet »Sektor/Block belegt« Byte 005 enthält die Belegung für Sektor 0–7 Byte 006 enthält die Belegung für Sektor 8–16 Byte 007 enthält die Belegung für Sektor 17–23 (Sektor 21–23 sind natürlich nie vorhanden)
008–011	s.o. 004–007 für Spur 2
...	
140–143	s.o. 004–007 für Spur 35
144–159	Diskettenname, der bei der Formatierung angegeben wird; aufgefüllt mit Charactercodes 160 ($a0)
160–163	zweimal 160 ($a0) »SHIFT SPACE«
164	160 ($a0) »SHIFT SPACE«
165–166	$32 und $41 "2A"; Formatangabe der Diskette
167–170	160 ($a0) »SHIFT SPACE«
171–179	$00 bei 1541-Modus; $a0 bei 1570/71-Modus
180–220	0 ($00); nicht benutzter Bereich
221–255	1541/1570: restlicher Bereich nicht verwendet.
bei 1571:	
221–237	Anzahl der freien Blöcke für Spur 36–52
238	Anzahl der freien Blöcke für Spur 53 (immer 0)
239–244	Anzahl der freien Blöcke für Spur 54–59
245–250	Anzahl der freien Blöcke für Spur 60–65
251–255	Anzahl der freien Blöcke für Spur 66–70

Tabelle 2. Aufbau und Inhalt der BAM (Block-Belegungs-Plan) in Spur 18, Sektor 0

Die 1571 enthält zusätzlich noch ein Verzeichnis in Block 53,0:

Byte	Bedeutung
000	enthält 0 ($00)
001–003	s.o. 005–007 für Spur 36
...	
102–104	s.o. 005–007 für Spur 70
105–255	restlicher Bereich nicht verwendet

Aufbau eines 4-Byte-Eintrages in der BAM (eine Spur)

BYTE(s)	Bedeutung:
000	Anzahl der freien Blöcke dieser Spur
001–003	Belegplan der Spur. Jedes Byte ist zuständig für 8 Sektoren: Byte 1 für 0–7 Bit 7 für Sektor 0 Bit 6 für Sektor 1 und so weiter Byte 2 für 8–15 Byte 3 für 16–23

Tabelle 3. Für jede Spur reserviert die BAM 4 Byte

Um auch die folgenden Seiten des Blocks zu betrachten, drücken Sie zum Vorwärtsblättern <F1>; die weitere Bedienung ist analog zur oben beschriebenen. Rückwärtsblättern ist durch Drücken von <F2> möglich.

Fahren Sie nun bis zum Byte 144 vor und sehen Sie sich die Seite an.

Die Byte 144 bis 161 enthalten den Namen der Diskette, der beim Formatieren festgelegt wird. Direkt im Anschluß daran folgen die Byte 162,163, die die ID im ASCII-Code beinhalten, gefolgt von einem »SHIFT SPACE«. An der ID erkennt die Floppystation, ob die Diskette gewechselt wurde; deshalb sollte jede Diskette eine andere ID haben.

Byte 165 und 166 enthalten DOS-Version und Formatkennzeichen, hier normalerweise »2A«, wiederum gefolgt von einem »Shift Space«.

Die Bytes 171 bis 255 haben normalerweise keine Bedeutung und können unterschiedlich gefüllt sein. Bei Geos-Disketten steht hier die Formatkennung »GEOS V1.0« an der dieses Betriebssystem »seine« Disketten erkennt.

Aufbau eines Directory-Blocks:	
BYTE(s)	**Bedeutung**
000-001	Spur und Sektor des nächsten Dir.-Blocks
002-031	Eintrag Nummer 1
032-033	unbenutzt
034-063	Eintrag Nummer 2
064-066	unbenutzt
067-225	Einträge Nrummer 3-7 beziehungsweise unbenutzt
226-255	Eintrag Nummer 8

Tabelle 4. Aufbau des Directory der 1541/70/71

Aufbau eines Directory-Eintrags:	
BYTE(s)	**Bedeutung**
000	Filetyp, siehe gesonderte Tabelle
001-002	Spur und Sektor des ersten Datenblocks
003-018	Filename, aufgefüllt mit Charactercode 160
019-020	REL-Files: Spur und Sektor des ersten Side-Sektor-Blocks
021	REL-Files: Datensatzlänge
022-025	unbenutzt
026-027	Spur und Sektor beim Überschreiben mit @ (nur Zwischenspeicher)
028-029	Anzahl der von diesem File belegten Blocks

Tabelle 5. Bedeutung der einzelnen Byte des Directory

Aufbau des Filetyp-Bytes					
BIT	**Bedeutung, in Klammern jeweiliger Inhalt**				
0	(0)	(1)	(0)	(1)	(0)
1	(0)=DEL	(0)=SEQ	(1)=PRG	(1)=USR	(0)=REL
2	(0)	(0)	(0)	(0)	(1)
3	unbenutzt				
4	unbenutzt				
5	unbenutzt				
6	(0)=normal; (1) = File kann durch SCRATCH nicht mehr gelöscht werden				
7	(0)= File noch offen (1)= File ordnungsgemäß geschlossen				

Tabelle 6. Die Bedeutung des ersten Bytes eines Directory-Eintrages

Aufbau eines Side-Sektor-Blocks:	
BYTE(s)	**Bedeutung**
000-001	Spur und Sektor des nächsten Side-Sektor-Blocks
002	Nummer des Side-Sektor-Blocks
003	Datensatzlänge
004-005	Spur und Sektor des Side-Sektor-Blocks 1
006-007	Spur und Sektor des Side-Sektor-Blocks 2
008-015	Spur und Sektor der Side-Sektor-Blöcke 3-6
016-017	Spur und Sektor des ersten Datenblocks, für den der Side-Sektor-Block zuständig ist (Datenblock 0) Spur und Sektor des zweiten Datenblocks (Nummer 1)
018-255	Spur und Sektor der Datenblocks Nummer 2 bis Nummer 119

Tabelle 7. Relative Dateien benutzen Side-Sektor-Blöcke, um Datensätze gezielt anzuspringen

Auf unserer Entdeckungsreise durch Spur 18 folgen wir jetzt der Angabe in den ersten beiden Byte und laden den ersten Directory-Block (<F3>; 18,1). Das Format des Blocks ist der Tabelle 4 zu entnehmen. Jeder Directory-Block enthält acht File-Einträge und den Zeiger auf den nächsten Directory-Block (Byte 0 und 1); ist die Track-Nummer des nächsten Blocks 0, so war der gelesene Directory-Block der letzte, und das zweite Byte zeigt die Anzahl der hier benutzten Byte. In unserem Fall stehen hier 0 und 255.

Nun zu Tabelle 5, die das Format eines Directory-Eintrags darlegt: Jeder dieser Einträge besteht aus 30 Byte, wobei das erste den Filetyp (siehe Tabelle 6), die beiden nächsten Spuren und Sektoren des ersten Fileblocks und die 16 folgenden Bytes den Filenamen enthalten. Die folgenden 3 Byte werden nur bei relativen Dateien verwendet; sie werden später im einzelnen noch besprochen.

Byte 26 und 27 enthalten Track und Sektor des neuen Files, falls das alte mit »@« überschrieben wurde. Byte 28 und 29 schließlich geben die Anzahl der belegten Blöcke dieses Files an.

Die einzelnen Datei-Typen

Diese bis jetzt beschriebenen Angaben werden vom Betriebssystem der Floppy, also vom DOS (englisch: Disk Operating System), verwaltet.

Beschäftigen wir uns nun mit den restlichen Blöcken auf der Diskette, die dem Anwender zur freien Verfügung stehen, denn dort werden die einzelnen Files gespeichert, deren Aufbau uns jetzt interessiert.

DEL-Files:

Diese Fileanzeige existiert normalerweise nicht im Directory; wird ein File gelöscht, so wird dieses nicht mehr angezeigt; das Byte des Filetyps steht dann auf 0. Durch Setzen des Filetyps auf 128 (hex. $80) kann eine DEL-Anzeige jedoch erzwungen werden.

SEQ-Files:

Dieser Filetyp dient zur Speicherung von Daten auf Diskette (im Gegensatz zur Programmspeicherung). Der Aufbau dieses Filetyps ist relativ einfach: Die ersten beiden Bytes eines Datenblocks zeigen jeweils auf den nächsten Block im File; so erfolgt eine beliebig lange Blockverkettung auf der Diskette. Da aber auch das schönste File einmal zu Ende geht, muß der letzte Block gekennzeichnet sein. Dies erfolgt, wie schon beim Directory, durch eine 0 als Spurnummer. Die Sektornummer bezeichnet jetzt die Anzahl der belegten Datenbytes dieses Blocks. Diese Art der Verkettung von Blöcken wird bei allen Filetypen vorgenommen! Die restlichen 254 Byte jedes Blocks enthalten die Daten.

USR-Files:

USR-Files stimmen im Aufbau exakt mit den SEQ-Files überein, sie haben jedoch noch Zusatzfunktionen im DOS, auf die später eingegangen werden soll.

PRG-Files:

PRG-Files stellen den häufigsten Filetyp dar. Sie dienen der Speicherung von Programmen auf der Diskette und haben nahezu denselben Aufbau wie SEQ-Files. Der einzige Unterschied besteht in den Byte 2 und 3 des ersten Blocks, welche die Startadresse des Programms im Computer enthalten. Ist diese Adresse gleich der Adresse des Basic-Anfangs, also 2049 ($0801), so können die Programme mit »LOAD"Name",8« geladen werden; dieser Modus ignoriert die Anfangsadresse auf Diskette und lädt die Programme generell an den Basic-Anfang (sogenanntes relatives La-

```
10 REM EDDI - DISKMONITOR/EDITOR                    <150>
50 PRINT"{CLR,BLACK}":POKE 53280,14:POKE 5
   3281,14                                          <079>
60 GOSUB 10000                                      <040>
70 OPEN 1,8,15,"I0":OPEN 2,8,2,"#"                  <106>
80 PRINT"{CLR,SPACE}E D D I{2SPACE}-{2SPAC
   E}HAUPTMENUE"                                    <069>
85 HE$="BYTE{6SPACE}DEC{3SPACE}HEX{3SPACE}
   BIN{8SPACE}ASC":POKE 650,128                     <106>
90 PRINT" EEEEEEEEEEEEEEEEEEEEEE"                   <008>
100 PRINT:PRINT:PRINT                               <225>
110 PRINT"(F1) - SCROLLING VORWAERTS":PRIN
    T                                               <147>
120 PRINT"(F2) - SCROLLING RUECKWAERTS":PR
    INT                                             <144>
130 PRINT"(F3) - BLOCK LESEN":PRINT                 <078>
140 PRINT"(F4) - BLOCK SCHREIBEN":PRINT             <132>
150 PRINT"(F5) - EDITOR EINSCHALTEN":PRINT          <182>
160 PRINT"(F6) - DISKETTE WECHSELN":PRINT           <094>
170 PRINT"(F7) - RUECKKEHR INS MENUE":PRIN
    T                                               <215>
180 PRINT"(F8) - PROGRAMMENDE"                      <037>
190 PO=1:GOTO 9000                                  <246>
1000 REM EDDI AN                                    <098>
1010 X=0:Y=0                                        <023>
1020 FOR Y=E TO 255 STEP 16                         <024>
1030 PO=2:PRINT"{CLR}EDITOR-MODUS FUER TRA
     CK"T" SEKTOR"S                                 <138>
1040 PRINT:PRINT HE$:PRINT                          <193>
1050 FOR X=Y TO Y+15:PRINT X:NEXT X                 <145>
1060 PRINT"{HOME,3DOWN}":FOR X=Y TO Y+15            <043>
1065 DA=PEEK(50000+X):GOSUB 7030:PRINT X,O
     U$                                             <208>
1070 INPUT"{UP,8RIGHT}";IN$:IF IN$=""THEN
     1090                                           <132>
1072 IF LEFT$(IN$,1)="↑"THEN PRINT"{HOME,1
     9DOWN}":GOTO 9000                              <207>
1073 IF LEFT$(IN$,1)="←"THEN PRINT"{HOME,2
     0DOWN}":GOTO 1125                              <196>
1075 DA=VAL(LEFT$(IN$,3)):IF DA>255 OR DA<
     0 THEN PRINT"{2UP}":GOTO 1065                  <228>
1080 POKE 50000+X,DA                                <180>
1120 NEXT X:PRINT                                   <010>
1125 PRINT"EINGABE ?";                              <243>
1130 GET A$:IF A$=""THEN 1130                       <228>
1140 IF A$="{F1}"THEN 1200                          <101>
1150 IF A$="{F2}"THEN 1300                          <121>
1160 IF A$<>" "THEN NEXT Y                          <066>
1170 PO=1:GOTO 9000                                 <210>
1200 PRINT"{HOME,3DOWN}":PRINT E".....???"          <179>
1210 GET A$:IF A$=""THEN 1210                       <115>
1215 IF A$="{F2}"THEN 1300                          <186>
1220 IF A$<>"{F1}"THEN 1020                         <126>
1230 E=E+16:IF E>255 THEN E=0                       <214>
1240 GOTO 1200                                      <196>
1300 PRINT"{HOME,3DOWN}":PRINT E".....???"          <025>
1310 GET A$:IF A$=""THEN 1310                       <025>
1315 IF A$="{F1}"THEN 1200                          <022>
1320 IF A$<>"{F2}"THEN 1020                         <232>
1330 E=E-16:IF E<0 THEN E=240                       <020>
1340 GOTO 1300                                      <058>
2000 REM DISKETTENWECHSEL                           <082>
2010 PRINT"{CLR}BITTE NEUE DISKETTE EINLEG
     EN"                                            <171>
2020 GET A$:IF A$=""THEN 2020                       <196>
2030 RUN                                            <038>
3000 REM BLOCK READ                                 <067>
3010 PO=2:PRINT"{CLR,3SPACE}BLOCK LESEN":P
     RINT:PRINT                                     <093>
3020 INPUT"TRACK, SEKTOR ";T,S                      <092>
3025 IF T<1 OR T>35 THEN 3010                       <051>
3030 PRINT#1,"U1 2 0"T;S                            <163>
3035 IF ST<>0 THEN PRINT:GOTO 9000                  <195>
3040 PRINT#1,"B-P 2 0"                              <146>
3050 SYS 49152:E=0:X=0:Y=0:GOTO 5010                <188>
3060 FOR Y=E TO 255 STEP 16                         <032>
3070 PRINT"{CLR}TRACK"T" SEKTOR"S                   <136>
3080 PRINT:PRINT HE$:PRINT                          <201>
3090 FOR X=Y TO Y+15:DA=PEEK(50000+X):GOSU
     B 7030:PRINT X,OU$:NEXT X                      <138>
3100 GOTO 9000                                      <056>
4000 REM BLOCK WRITE                                <070>
4010 PO=1:PRINT:PRINT:INPUT"{CLR,RED}TRACK
     , SEKTOR";T,S:PRINT"{BLACK}"                   <169>
4020 PRINT#1,"B-P 2 0"                              <110>
4030 SYS 49177                                      <024>
4040 PRINT#1,"U2 2 0"T;S                            <030>
4050 GOTO 9000                                      <244>
5000 REM SCROLL FORWARD                             <247>
5010 E=X:IF E>255 THEN X=0:E=0                      <156>
5020 PRINT"{CLR}TRACK"T" SEKTOR"S                   <054>
5030 PRINT:PRINT HE$:PRINT                          <117>
5040 DA=PEEK(50000+E):GOSUB 7030:PRINT E,O
     U$                                             <136>
5050 X=X+16                                         <023>
5060 GET A$:IF A$=""THEN 5060                       <030>
5070 IF A$="{F1}"THEN 5010                          <237>
5075 IF A$="{F2}"THEN X=X-16:GOTO 6010              <156>
5077 IF A$="{F5}"THEN 1000                          <213>
5080 GOTO 3060                                      <146>
6000 REM SCROLL BACKWARD                            <055>
6010 E=X:IF E<0 THEN E=240:X=240                    <161>
6020 PRINT"{CLR}TRACK"T" SEKTOR"S                   <038>
6030 PRINT:PRINT HE$:PRINT                          <101>
6040 DA=PEEK(50000+E):GOSUB 7030:PRINT E,O
     U$                                             <120>
6050 X=X-16                                         <039>
6060 GET A$:IF A$=""THEN 6060                       <046>
6070 IF A$="{F2}"THEN 6010                          <227>
6075 IF A$="{F1}"THEN X=X+16:GOTO 5010              <102>
6077 IF A$="{F5}"THEN 1000                          <197>
6080 GOTO 3060                                      <130>
7000 REM BEREITSTELLUNG DES STRINGS                 <077>
7010 REM DA/DA$ SIND AUSGABEWERTE
          H$,D$,B$,C$ SIND ZWISCHENWERTE            <087>
7020 REM OU,OU$ SIND ENDERGEBNISSE                  <168>
7030 IF DA>31 AND DA<128 OR DA>159 AND DA<
     256 THEN C$=CHR$(DA):GOTO 7040                 <166>
7035 C$="."                                         <234>
7040 XX$="000":D$=RIGHT$(STR$(DA),LEN(STR$
     (DA))-1)                                       <056>
7045 D$=LEFT$(XX$,3-LEN(D$))+D$                     <229>
7050 XX$="123456789ABCDEF":H$=""                    <225>
7060 HH=INT(DA/16):HL=DA-HH*16                      <126>
7070 IF HH THEN H$=H$+MID$(XX$,HH,1):GOTO
     7080                                           <163>
7075 H$=H$+"0"                                      <183>
7080 IF HL THEN H$=H$+MID$(XX$,HL,1):GOTO
     7090                                           <021>
7085 H$=H$+"0"                                      <193>
7090 B$="":FOR Q=7 TO 0 STEP-1                      <140>
7100 IF(DA AND(2↑Q))<>0 THEN B$=B$+"1":NEX
     T:GOTO 7110                                    <099>
7105 B$=B$+"0":NEXT                                 <245>
7110 OU$=D$+"{3SPACE}"+H$+"{4SPACE}"+B$+"{
     3SPACE}"+C$                                    <035>
7120 RETURN                                         <064>
8999 END                                            <111>
9000 REM GET KOMMANDO                               <200>
9010 PRINT:PRINT"KOMMANDO ?{2SPACE}";               <171>
9020 PRINT"{LEFT}▒";:FOR W=1 TO 75:GET KO$
     :IF KO$<>""THEN 9090                           <158>
9030 NEXT W                                         <080>
9040 PRINT"{LEFT,RVSON}▒{RVOFF}";:FOR W=1
     TO 75:GET KO$:IF KO$<>""THEN 9090              <220>
9050 NEXT W                                         <100>
9060 GOTO 9020                                      <238>
9090 IF KO$="@"THEN 9200                            <165>
9100 IF ASC(KO$)>140 OR ASC(KO$)<133 THEN
     9020                                           <202>
9110 KO=ASC(KO$)-132                                <210>
9120 ON PO GOTO 9130,9140,20000                     <086>
9130 ON KO GOTO 9020,3000,1000,80,9020,400
     0,2000,20000                                   <251>
9140 ON KO GOTO 5000,3000,1000,80,6000,400
     0,2000,20000                                   <247>
9200 PRINT                                          <156>
9210 GET#1,A$:PRINT A$;:IF ST<>64 THEN 921
     0                                              <082>
9220 GOTO 9000                                      <080>
9999 END                                            <095>
10000 DATA 160,0,169,8,32,9,237,169,98,32,
      199,237,32,19,238,153,80,195,200              <054>
10010 DATA 208,247,32,239,237,96,160,0,169
      ,8,32,12,237,169,98,32,185,237                <155>
10020 DATA 185,80,195,32,221,237,200,208,2
      47,32,254,237,96,0,0                          <054>
10030 RESTORE:FOR Z=1 TO 51:READ A:POKE 49
      151+Z,A:NEXT                                  <168>
10040 REM GET:49152; WRITE:49177                    <225>
10050 RETURN                                        <202>
20000 PRINT:PRINT:PRINT"{LIG.BLUE}AUF WIED
      ERSEHEN !!!":PRINT:POKE 53280,14:POK
      E 53281,6                                     <099>
20001 PRINT"UND DANKESCHOEN !"                      <133>
```

Listing 1. EDDI, ein Disk-Monitor/Editor

den). Sollen Programme jedoch an anderen Stellen im Speicher stehen, zum Beispiel Maschinenprogramme, so muß diese angegebene Adresse als Startadresse benutzt werden; man lädt hier absolut mit »LOAD"Name",8,1«.

REL-Files:

Dieser Filetyp ist im Aufbau ungleich komplizierter als die eben besprochenen; es soll daher zuerst kurz auf die Arbeitsweise von REL-Files eingegangen werden. Sequentielle Files haben den Nachteil, daß sie praktisch nur aus einem Datensatz bestehen. Sucht man nun, zum Beispiel in einer Kartei, eine bestimmte Hausnummer oder einen bestimmten Namen, so muß der gesamte Datensatz durchgelesen werden, um die entsprechende Stelle zu finden. In einer relativen Datei geht man deshalb einen anderen Weg, um jede Stelle schnell auffinden zu können.

Es existiert eine beliebige Anzahl (zum Beispiel 100) von Datensätzen, wobei alle Datensätze die gleiche Länge haben müssen (maximal 254 Zeichen).

Das DOS legt jetzt einen sogenannten Side-Sektor an, der aus bis zu sechs Blöcken bestehen kann. Diese Blöcke enthalten die Zeiger auf sämtliche Datenblöcke, in denen die Datensätze gespeichert sind (1 Datensatz hat maximal 1 Block Länge). Auch hier sind die Datenblöcke wieder durch Zeiger in den Byte 0 und 1 verkettet. Den Aufbau eines Side-Sektor-Blocks zeigt Tabelle 7. Zum besseren Verständnis hier ein kleines Beispiel:

Wir haben eine relative Datei mit 250 Datensätzen à 127 Zeichen. Diese Datei benötigt also 125 Datenblöcke und zwei Side-Sektor-Blöcke. Im Directory-Eintrag finden wir jetzt die schon erwähnten zusätzlichen Byte-Belegungen: Byte 19 und 20 jedes Eintrags enthalten jetzt Spur und Sektor des ersten Side-Sektor-Blocks; Byte 21 gibt die Datensatzlänge (Recordlänge) an.

Wir wollen jetzt auf den 248. Datensatz zugreifen; das DOS arbeitet nun folgendermaßen: Ein Datensatz enthält 127 Byte, das heißt, es passen zwei Datensätze in einen Block; dadurch errechnet sich der Block, auf den jetzt zugegriffen wird, aus (248-1)/2=123,5. (Minus 1, da immer von 0 an gezählt wird.) Da ein Side-Sektor-Block nur 120 Einträge aufnehmen kann, ist der Zeiger auf den Datenblock im Side-Sektor-Block Nummer 2 zu finden.

Dieser wird jetzt anhand des Verzeichnisses in Block 1 gelesen und dann auf Zeiger Nummer 3 (Byte 22,23) zugegriffen. Wir kennen also jetzt Spur und Sektor des Blocks, in dem unser Datensatz steht; die Position des ersten Daten-Byte berechnet sich jetzt aus dem Nachkommaanteil der obigen Division (0.5x254=127). Der Datensatz beginnt also beim 127+2=129. Byte.

Der Aufbau von relativen Dateien ist also, wie schon erwähnt, ziemlich kompliziert; diese Art der Datenspeicherung hat aber einige Vorteile gegenüber der »normalen« mit SEQ-Files.

Da unserem U-Boot auf dieser schwierigen Fahrt der Sauerstoff ausgegangen ist, wollen wir uns nun erst einmal erholen. Hier noch ein paar Anregungen zur Arbeit mit EDDI: EDDI kann nicht nur Blöcke lesen und anzeigen; Sie können auch Bytes verändern und diesen Block danach wieder speichern.

Bedienung von EDDI

Dazu laden Sie den zu verändernden Block und wählen die Seite, die Sie interessiert an; hier tippen Sie als Kommando <F5>, und der Editormodus startet. Sie können jetzt Bytes dezimal abändern, indem Sie den jeweils neuen Wert eingeben und <RETURN> drücken. Wollen Sie aus dem Eingabemodus aussteigen, so tippen Sie entweder <RETURN> und können weiterblättern, ohne den Editor zu verlassen, oder Sie tippen <↑><RETURN>, um in den Kommandomodus zu kommen. Nach einigem Probieren wird Ihnen EDDI sehr schnell vertraut werden; wir gehen auch in den folgenden Abschnitten noch darauf ein.

Wichtig:

Beim Wechseln einer Diskette muß die Funktionstaste <F6> betätigt und nach dem Austausch eine Taste gedrückt werden, sonst reagiert das Laufwerk mit einer Fehlermeldung. Diese können übrigens mit <@> abgerufen werden. Das Zurückschreiben eines Blocks auf Diskette erfolgt mit <F4>, wobei Spur und Sektornummer angegeben werden müssen. Hier noch ein paar Vorschläge zum Ausprobieren: Ändern Sie doch einmal auf Ihrer Versuchsdiskette (!) das Formatkennzeichen (Spur 18, Sektor 0, Byte 2 auf 66 statt jetzt 65) und speichern den Block an die gleiche Stelle auf die Diskette zurück. Versuchen Sie nun einmal, ein kleines Programm auf dieser Diskette zu speichern. (Die genauen Vorgänge in der Floppy-Station werden später erläutert.) Oder ändern Sie einmal die Bytes im Directory, die den Filetyp angeben, entsprechend Tabelle 6 und laden Sie es danach. Experimentieren Sie ruhig ein wenig mit der Floppy-Station. Das wird Ihnen das Verständnis im weiteren Verlauf des Kurses stark erleichten.

Floppy kontra Datasette

Sicherlich machte sich jeder, der ein schnelleres Peripheriegerät als die Datasette haben wollte, schon seine Gedanken über den Preis der 1541/70/71: »Die kostet ja mehr als der Computer!« In der Tat ist die 1541-Floppystation von dieser Seite her betrachtet nicht gerade günstig, wer sich jedoch schon intensiver mit ihr beschäftigt hat, wird eine Eigenart festgestellt haben, die sie mit allen anderen Commodore-Laufwerken teilt: Sie ist »intelligent«. Diese Laufwerke besitzen ein eigenes Betriebssystem (DOS) und eigene Mikroprozessoren. Sie arbeiten völlig unabhängig vom Computer und dessen Speicher. Der Vorteil liegt auf der Hand: Das 1541-Laufwerk beansprucht weder Speicherplatz noch Rechenzeit des Computers, außer beim direkten Datenaustausch. Als Beispiel betrachte man den Befehl »N« (Formatieren). Während der Formatierung steht der Computer zur (fast) freien Verfügung, da dieser Vorgang nur laufwerksintern abläuft und sich der C 64 mit READY meldet, während die 1541 noch arbeitet.

Wir wollen uns jedoch nur den Direktzugriffsbefehlen und den Speicherbefehlen widmen; auch übergehen wir die im Commodore-Handbuch nicht erwähnte relative Datenspeicherung, über die in anderen Ausgaben schon ausführlich gesprochen wurde. Uns sollen nur die Befehle beschäftigen, die uns zur willkürlichen Manipulation von Floppystation und Disketten nützen.

Zur Beruhigung: Ein Beschädigen der 1541 durch direkte Eingriffe in das DOS ist nicht zu befürchten, auch wenn es passieren kann, daß sich das Laufwerk nur noch durch Aus-/Einschalten wieder in den Normalzustand versetzen läßt. Haben Sie übrigens einmal, wie empfohlen, das Formatkennzeichen einer Diskette verändert? Sie werden sicherlich bemerkt haben, daß sich danach nichts mehr auf Ihre Diskette schreiben läßt. Mit diesem Trick, der die gleichen Folgen wie das Anbringen einer Schreibschutzplakette an der Diskette hat, können Sie sich also ganz einfach Ihre Diskette gegen unbeabsichtigtes Löschen sichern. ACHTUNG: Diese Methode funktioniert natürlich nicht, wenn neu formatiert werden soll; dagegen hilft nur das Anbringen einer Schreibschutzplakette!

Die Floppystation verfügt, außer den schon bekannten Befehlen zur Diskettenorganisation, über eine ganze Anzahl weiterer Befehle, mit denen sich ungeahnte Möglich-

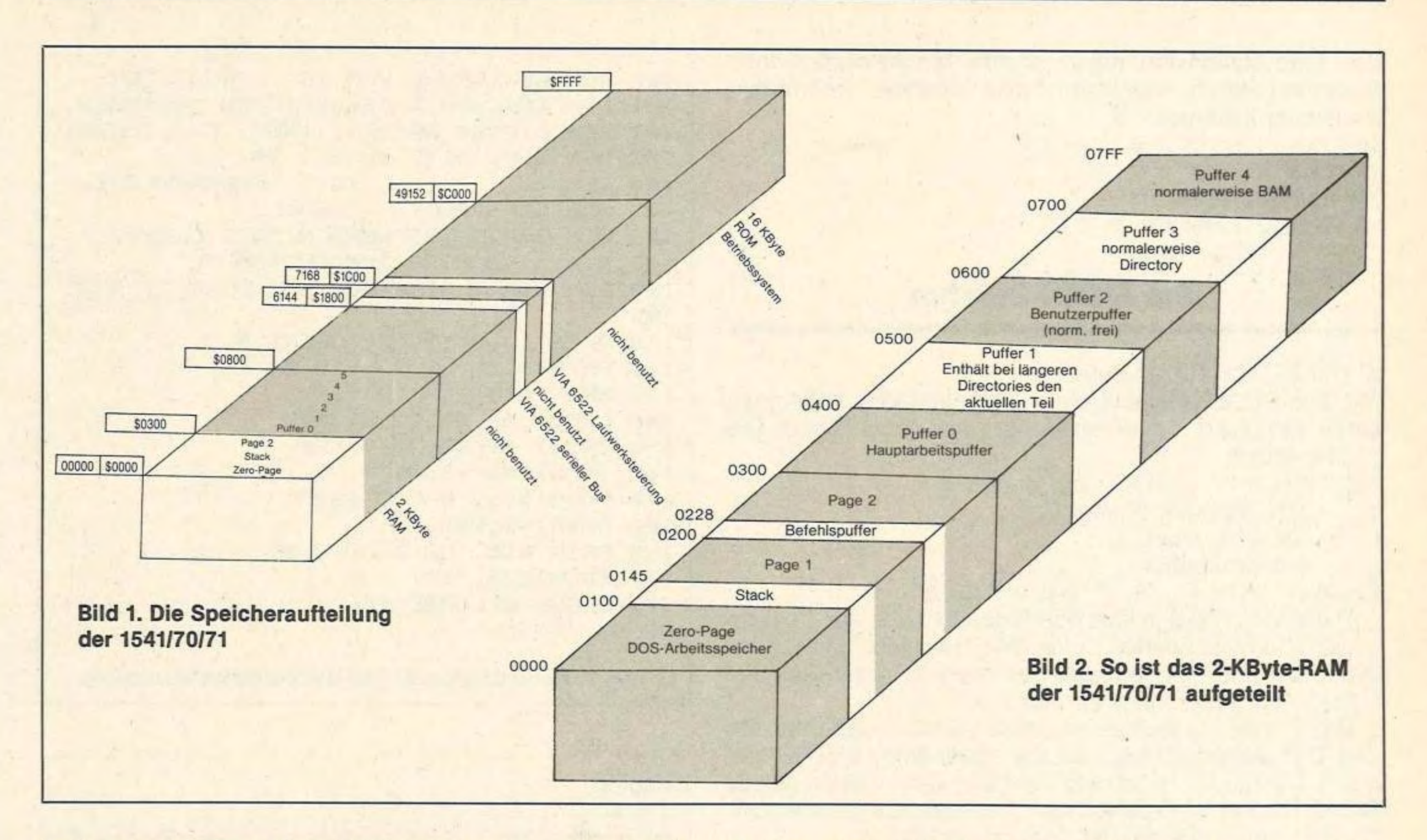


Bild 1. Die Speicheraufteilung der 1541/70/71

Bild 2. So ist das 2-KByte-RAM der 1541/70/71 aufgeteilt

keiten ergeben, zum Beispiel Herstellen eines eigenen Diskettenformats, Leseschutz von Disketten, Programmschutz, Modifikation der Lade- und Saveroutinen und, und, und. Dafür ist es allerdings nötig, daß wir diese Befehle Schritt für Schritt kennenlernen, bevor wir auf die Tricks der Profis, die Manipulationen des DOS und den gezielten Eingriff in den Programmablauf der Floppystation zu sprechen kommen. Dafür ist allerdings das Beherrschen des C 64 und der Maschinensprache unerläßlich. So lohnt es sich unter Umständen, nachdem man aus dem Basic nichts mehr herausholen kann, den Einstieg in die Assemblerprogrammierung zu wagen. Sehr gute Literatur dafür ist vorhanden. Aber vorerst wollen wir uns noch auf das Basic beschränken, um Sie mit dem Befehlssatz der Floppystation vertraut zu machen.

Befehle an die 1541/70/71

Wie schon erwähnt, handelt es sich bei der 1541 um einen vollständigen Computer, der ebenso wie Ihr C 64 RAM und ein Betriebssystem (DOS) im ROM besitzt.

Die genaue Aufteilung ist in Bild 1 zu sehen. Jetzt soll uns nur der RAM-Bereich interessieren (Bild 2). Nicht nur auf der Diskette, sondern auch im RAM werden Speicherbereiche in Abschnitte zu jeweils 256 Byte aufgeteilt. Sie heißen dann nicht mehr Blocks, sondern Pages (Seiten). Das RAM der 1541/70/71 umfaßt nun genau 8 Pages, durchnumeriert von 0 bis 7, insgesamt also 2 KByte. Die Page Nummer 0 (auch Zero-Page genannt) wird hier, wie auch im C 64, vom Betriebssystem als Arbeitsspeicher benutzt und steht uns deshalb nicht zur freien Verfügung. Ähnlich verhält es sich mit den Pages 1 und 2. Die Pages 3 bis 7 stellen sogenannte Pufferspeicher dar; hier werden alle Daten, die von der Diskette gelesen beziehungsweise auf sie geschrieben werden, zwischengespeichert, da nur blockweise gelesen oder geschrieben werden kann.

Soll zum Beispiel nur ein einziges Byte auf der Diskette geändert werden, so wird erst der gesamte Block in einen der 5 Pufferspeicher gelesen, dort abgeändert und schließlich komplett wieder zurückgeschrieben. Aus diesen Gründen ist es also notwendig, daß wir uns vor einem Direktzugriff einen der Puffer reservieren, in dem dann gearbeitet wird.

Mit Hilfe des »OPEN«-Befehls eröffnen wir einen Direktzugriffskanal. Die Syntax lautet wie folgt:

```
OPEN fn, gn, kn, "#"
```

Hierbei bedeuten:
fn - Filenummer (1-127)
gn - Gerätenummer (normalerweise 8)
kn - Kanalnummer in der Floppy (2-14)

Diese Abkürzungen werden wir im folgenden immer verwenden! Ein Beispiel:

```
OPEN 1, 8, 2, #
```

Diese Anweisung öffnet im Computer ein File mit der Nummer 1, adressiert als Gerät die Floppystation (Nummer 8) und reserviert in der 1541 einen Kanal (Nummer 2), dem ein Puffer zugeordnet wird. Mit den laufwerksinternen Kanälen verhält es sich wie folgt: Es stehen insgesamt 16 Kanäle zur Verfügung. Hierbei sind Kanal 0 und 1 für LOAD und SAVE reserviert, Kanal 15 ist der Kommandokanal, den Sie bisher immer benutzt haben, um Befehle (zum Beispiel Formatieren) an das Laufwerk zu senden und die Fehlermeldungen des Laufwerks zu empfangen.

Für unsere Zwecke stehen also noch die Kanäle 2 bis 14 zur Verfügung. In unserem Fall reserviert die Floppystation den nächsten freien Puffer. Will man jedoch einen bestimmten Puffer reservieren, etwa um dort ein Maschinenprogramm abzulegen, so ist es notwendig, der 1541 mitzuteilen, welcher Puffer gewünscht wird:

```
OPEN 1, 8, 2, "#1"
```

Es ist hier allerdings zu beachten, daß der gewählte Puffer nicht schon belegt ist; in diesem Fall gibt die 1541 eine Fehlermeldung aus. Wollen Sie an dieser Stelle mehr über das Auslesen der Fehlermeldungen und deren Bedeutung wissen, können wir Sie hier beruhigt auf das Commodore-Handbuch verweisen, das hier sehr ausführlich ist.

Im allgemeinen sind Puffer 4 für die BAM und Puffer 3 für das Directory reserviert. Haben Sie die Wahl des Puffers

der Floppystation überlassen, so erfahren Sie die gewählte Nummer durch Auslesen des soeben geöffneten Direktzugriffskanals:

```
10 OPEN 1, 8, 2, "#"
20 GET# 1, D$
30 D=ASC(D$+CHR$(0))
40 REM PUFFERNUMMER IN D
```

Die BLOCK-Befehle

a) Der BLOCK-READ-Befehl (B-R):
Mit dem BLOCK-READ-Befehl liest man jeden beliebigen Block von Diskette in einen vorher reservierten Puffer. Die Syntax lautet:

`PRINT# fn, " B-R" ; kn; dn; t; s`

dn - Drivenummer (immer 0)
t - Track-Nummer
s - Sektornummer

Beispiel: `PRINT# 15, " B-R 2 0 18 0"`

Diese Befehlsfolge liest den Block 18,0 von der Diskette in den oben reservierten Puffer. Wie man sieht, können anstelle der CHR$-Codes feste Zahlenwerte in den Befehlsstring mit übernommen werden.

Das Ganze hat bloß einen kleinen Schönheitsfehler. Mit dem B-R-Befehl läßt sich das erste Byte eines Blocks nicht lesen. Deshalb benutzt man normalerweise anstatt des B-R-Befehls den U1-Befehl. Dieser hat exakt die gleiche Syntax und kann in jedem Fall benutzt werden:

`PRINT# 15, " U1 2 0 18 0"`

Auf diese User-Befehle kommen wir später zurück. Mit einer GET#-Schleife lassen sich nun die einzelnen Bytes in den Computer einlesen.

b) Der BLOCK-WRITE-Befehl (B-W):
Hiermit lassen sich die Daten aus dem reservierten Puffer wieder auf die Diskette schreiben. Syntax:

`PRINT# fn, " B-W" ; kn; dn; t; s`

Beispiel: `PRINT# 15, " B-W 2 0 18 0"`

Natürlich gibt es analog zum B-W- einen USER-Befehl: U2. Beispiel:

`PRINT# 15, " U2 2 0 18 0" .`

c) Der BUFFER-POINTER-Befehl (B-P):
Für jeden Puffer gibt es einen Zeiger, den Buffer-Pointer. Dieser zeigt auf das aktuelle Byte im Puffer und wird bei jedem Datenzugriff um eins erhöht, damit man alle 256 Byte eines Blocks der Reihe nach lesen kann.

Dieser Pointer wird mit dem B-P-Befehl gezielt auf bestimmte Bytes positioniert, wenn man nur einzelne Werte und nicht den gesamten Block lesen will.

Syntax:

`PRINT# fn, " B-P" ; kn; position`

Beispiel:
Wir möchten in die Variable A den Wert des 123. Bytes von Block 1;16 einlesen:

```
10 OPEN 15, 8, 15
20 OPEN 1, 8, 2," #  "
30 PRINT# 15, " U1 2 0 1 16"
40 PRINT# 15, " B-P 2 122"
50 GET# 1, A$
60 A=ASC(A$+CHR$(0))
```

Als weiteres Beispiel dient Listing 1.

d) Der BLOCK-ALLOCATE-Befehl (B-A):
Wenn Sie im Direktzugriffsverfahren eine Diskette beschreiben, muß in der BAM danach auch verzeichnet werden, daß die entsprechenden Blocks mit Daten gefüllt sind und nicht mehr überschrieben werden dürfen. Dazu dient der B-A-Befehl, der jeden beliebigen Block in der BAM als belegt kennzeichnet. Die Syntax lautet:

`PRINT# fn, " B-A" ; dn; t; s`

```
100 REM AENDERUNG VON ID, FORMATKENN-
101 REM ZEICHEN & LEERZEICHEN ZWISCHEN
102 REM DIESEN BEIDEN.(INSG. 5 ZEICHEN)
103 REM BSP: ALTE ID :XY 2A
104 REM              ID^  ^FORMATKENNZ.
105 REM KANN AUF         :HALLO
106 REM GEAENDERT WERDEN. DAS LEERZ.
107 REM WIRD HIER ZUM ERSTEN 'L'
108 REM WIRKT SICH NUR AUF DIRECT. AUS!
109 :
110 OPEN 15,8,15,"I":OPEN1,8,2,"#"
120 PRINT#15,"U1 2 0 18 0"
130 PRINT#15,"B-P 2 162"
140 GET#1,A$,B$,C$,D$,E$
150 PRINT A$;B$;C$;D$;E$
160 INPUT"NEU:";N$
170 PRINT#15,"B-P 2 162"
180 PRINT#1,N$;
190 PRINT#15,"U2 2 0 18 0"
200 PRINT#15,"I"
210 CLOSE 8:CLOSE 15
READY.
```

Listing 2. Änderung der ID und des Formatkennzeichens

Beispiel:

`PRINT# 15, " B-A 0 1 16"`

kennzeichnet Block 1;16 als belegt; war dieser Block schon belegt, meldet sich die Floppystation mit der Fehlermeldung »65, NO BLOCK, XX, YY«; wobei XX und YY die Track- und Sektornummer des nächsten freien Blocks angeben.

e) Der BLOCK-FREE-Befehl (B-F):
Dieser ist das genaue Gegenstück zum B-A-Befehl; er deklariert einmal belegte Blöcke wieder als frei für einen weiteren Zugriff. Seine Syntax ist identisch mit der des B-A-Befehls.

f) Der BLOCK-EXECUTE-Befehl (B-E):
Dieser Befehl nimmt eine Sonderstellung ein. Er gleicht im Prinzip dem B-R-Befehl; nur mit dem zusätzlichen Effekt, daß der eingelesene Block im Puffer als Maschinenprogramm gestartet wird.

Zur Vertiefung der Block-Befehle sei noch auf die Listings 2 bis 7 hingewiesen, welche die eben besprochenen Anwendungen noch an praktischen Beispielen verdeutlichen.

Die MEMORY-Befehle

a) Der MEMORY-READ-Befehl (M-R):
Dieser Befehl entspricht haargenau dem PEEK-Befehl in Basic. Mit ihm können Sie jede beliebige Speicherstelle der Floppystation auslesen.

Syntax:

`PRINT#fn, " M-R" ; CHR$(adl); CHR$(adh); CHR$(n)`

adl = Low-Byte
adh = High-Byte
n = Anzahl (0 bis 255)

Abgeholt werden die gelesenen Daten ebenfalls über den Kommandokanal mit GET#.

Beispiel: Lesen der beiden ID-Zeichen im ASCII-Format der zuletzt initialisierten Diskette:

```
10 OPEN 15, 8, 15
20 PRINT# 15, " M-R"  CHR$(18) CHR$(0) CHR$(2)
30 GET# 15, A$, B$
40 PRINT A$; B$
```

Diese Routine liest die Zero-Page-Adressen 18 und 19, in denen die entsprechenden Werte gespeichert sind.

b) Der MEMORY-WRITE-Befehl (M-W):
Dieses Kommando kann als POKE-Befehl in den Floppyspeicher angesehen werden.
Die Syntax ist hier wie folgt:
`PRINT# fn, " M-W" ; CHR$(adl) CHR$(adh) CHR$(n) CHR$(data1) CHR$(data2)...`
c) Der MEMORY-EXECUTE-Befehl (M-E):
Auch dieser Befehl ist äquivalent zu einem Basic-Befehl, dem SYS-Befehl. Mit ihm kann man also ein Maschinenprogramm an einer beliebigen Stelle im Floppyspeicher ausführen. Syntax:
`PRINT# fn, " M-E" CHR$(adl) CHR$ (adh).`
Siehe auch Listing 8.

Die User-Befehle

Die User-Befehle stellen eine Erweiterung des Befehlssatzes dar, der fast ausschließlich der Bequemlichkeit dient. U1 und U2 wurden schon besprochen, sie ersetzen B-R und B-W.

Die Befehle U3 bis U8 dienen zum Starten eines Maschinenprogramms im Floppyspeicher, dessen Anfangsadressen in einer Tabelle abgelegt sind, so entsprechen:
U3 einem Start bei $0500
U4 einem Start bei $0503
...
U8 einem Start bei $050F.
U4 ersetzt also beispielsweise den Befehlsstring: M-E CHR$(3)CHR$(5).

U9 zeigt auf den NMI-Vektor der 1541, welcher allerdings eine Sonderfunktion hat: Mit U9+ wird die Floppystation auf C 64- und mit U9– auf VC 20-Betrieb umgeschaltet.
U: stellt einen Reset dar, ähnlich dem SYS 64738 beim C 64.

Mit den Kenntnissen über den Befehlssatz der 1541 dürfte es Ihnen nun keine Schwierigkeiten mehr bereiten, sich das Programm EDDI einmal zu Gemüte zu führen. Das einzig Besondere daran sind die Routinen zum Lesen und Schreiben eines Blocks, die aus Geschwindigkeitsgründen in Maschinensprache geschrieben sind.

1541/70/71 und Assembler

Wenn wir im folgenden von Routinen sprechen, die im Betriebssystem stehen, so werden wir die in Tabelle 8 dargestellten Kürzel verwenden, die Sie übrigens auch in Editorprogrammen gut benutzen können.

FILPAR und FILNAM

Bei OPEN, LOAD und ähnlichen Befehlen müssen Sie den entsprechenden Routinen mitteilen, welches File Sie wo öffnen wollen. Um Ihnen eine »Herumwurstelei« in der Zero-Page zu ersparen, wo Sie die einzelnen Angaben von Hand setzen müßten, hat das Betriebssystem zwei entsprechende Routinen implementiert. FILPAR setzt für Sie die einzelnen Fileparameter. Diese müssen der Routine in den Prozessorregistern übergeben werden:
- Filenummer (Akku)
- Geräteadresse (X-Register)
- Sekundäradresse (Y-Register)

Ein Beispiel:
Sie wollen für ein File mit der Nummer 1, der Geräteadresse 8 und der Sekundäradresse 15 (Kommandokanal) die entsprechenden Fileparameter setzen:

```
LDA   #  $01      ; Filenummer 1
LDX   #  $08      ; Geräteadresse 8
LDY   #  $6F      ; Sekundäradresse + $60
JSR   #  FILPAR   ; Fileparameter setzen
```

Wie Sie sehen, muß zu der betreffenden Sekundäradresse ein Wert von $60 addiert werden.

Aber in vielen Fällen müssen Sie ja auch einen Filenamen angeben. Dazu dient die FILNAM-Routine. Hier erfolgt die Parameterübergabe:

```
1000 REM UNTERPROGRAMM 1
1001 REM LESEN EINES EINTRAGES AUS DEM
1002 REM DIRECTORY (ALLE 30 BYTES !!!)
1003 REM IN DIE VARIABLE DD$
1004 REM UEBERGABEPARAMETER:
1005 REM MM=NUMMER DES EINTRAGES DER
1006 REM    GELESEN WERDEN SOLL
1007 :
1008 :
1009 :
1010 OPEN 15,8,15,"I":OPEN8,8,8,"#"
1020 NN$="":FORI=1TO30:NN$=NN$+CHR$(0):N
EXTI
1030 XX=INT((MM-1)/8)
1040 PRINT#15,"U1 8 0 18 0"
1050 FORZZ=1TOXX+1
1060 PRINT#15,"B-P 8 0"
1070 GET#8,TT$:TT=ASC(TT$+CHR$(0))
1080 GET#8,SS$:SS=ASC(SS$+CHR$(0))
1090 IF TT=0 THEN DD$=NN$:GOTO1170
1100 PRINT#15,"U1 8 0";TT;SS
1110 NEXTZZ
1120 PP=MM-(XX*8):PP=(PP-1)*32+2
1130 PRINT#15,"B-P 8";PP
1140 FORZZ=1 TO 30:GET#8,ZZ$
1150 IFZZ$=""THENZZ$=CHR$(0)
1160 DD$=DD$+ZZ$:NEXTZZ
1170 CLOSE 8:CLOSE 15
1180 RETURN
READY.
```

Listing 3. Unterprogramm 1. Lesen eines Eintrages aus dem Directory.

```
2000 REM UNTERPROGRAMM 2
2001 REM SCHREIBEN EINES EINTRAGES IN
2002 REM DAS DIRECTORY (30 BYTES !!!)
2003 REM UEBERGABEPARAMETER:
2004 REM MM=NUMMER DES EINTRAGES DER
2005 REM    GESCHRIEBEN WERDEN SOLL
2006 REM DD$=DIRECTORYEINTRAG
2007 :
2008 :
2009 :
2010 OPEN 15,8,15,"I":OPEN8,8,8,"#"
2020 XX=INT((MM-1)/8)
2030 PRINT#15,"U1 8 0 18 0"
2040 FORZZ=1TOXX+1
2050 PRINT#15,"B-P 8 0"
2060 GET#8,T$:TT=ASC(T$+CHR$(0))
2070 GET#8,S$:SS=ASC(S$+CHR$(0))
2080 IF TT=0 THEN 2150
2090 PRINT#15,"U1 8 0";TT;SS
2100 NEXTZZ
2110 PP=MM-(XX*8):PP=(PP-1)*32+2
2120 PRINT#15,"B-P 8";PP
2130 PRINT#8,DD$;
2140 PRINT#15,"U2 8 0";TT;SS
2150 CLOSE 8:CLOSE 15
2160 RETURN
READY.
```

Listing 4. Unterprogramm 2. Schreiben eines Directory-Eintrages

```
100 REM BEISPIEL FUER EINE KLEINE
101 REM DIRECTORY-MANIPULATION:
102 REM SCRATCH-SCHUTZ EINZELNER FILES
103 REM NACH ANZEIGE DES FILENAMENS:
104 REM J = SCHUETZE DIES FILE
105 REM N = WEITER ZUM NAECHSTEN FILE
106 REM E = ENDE
107 REM ACHTUNG !!! "SCHUETZT" AUCH
108 REM SCHON GESCRATCHTE FILES WENN
109 REM VERLANGT, STELLT SIE ABER NICHT
110 REM WIEDER HER !!!
111 REM SCRATCH-SCHUTZ WIRD IM DIRECT.
112 REM DURCH EIN '<' HINTER DEM
113 REM FILETYP ANGEZEIGT. NAEHERES
114 REM SIEHE TABELLE FOLGE 1 !!!
115 REM ACHTUNG !!! NUR ZUSAMMEN MIT
116 REM DEN UNTERPROGRAMMEN 1 & 2
117 REM LAUFFAEHIG !!!
118 :
119 :
120 MM=0
130 MM=MM+1:DD$="":GOSUB1000
140 IF DD$=NN$THENEND
150 PRINTMID$(DD$,4,16):INPUTAA$
160 IF AA$="E"THEN END
170 IF AA$="N"THEN 130
180 HH$=LEFT$(DD$,1)
190 HH$=CHR$(ASC(HH$)OR2^6)
200 DD$=HH$+RIGHT$(DD$,29)
210 GOSUB2000
220 GOTO 130
230 END
READY.
```

Listing 5. So kann man Files schützen

- Länge des Filenamens (Akku)
- Adresse LO des Namens (X-Register)
- Adresse HI des Namens (Y-Register)

Und wieder ein Beispiel. Um das Directory-File mit dem Namen »$« anzusprechen, geben Sie bitte folgende Befehle ein:

```
LDA   #$24     ; Code für '$' in Akku
STA    $FF     ; und abspeichern
LDA   #$01    ; Länge des Filenamens
LDX   #FF     ; Adresse LO
LDY   #$00    ; Adresse HI
JSR   FILNAM  ; übergeben
```

Sie müssen also wissen, wo der Filename im Speicher steht und wie lang er ist. Dies ist aber im allgemeinen kein Problem. Auf die gleiche Weise können Sie der Floppystation über den Kommandokanal auch Befehle senden, wie Sie in der letzten Folge vorgestellt wurden. Das entspräche der Basic-Sequenz:

OPEN x, 8, 15, "befehl"

Natürlich können Sie auch alle Parameter von Hand setzen, beziehungsweise noch einmal lesen. Wo sich die einzelnen Parameter in der Zero-Page nach Ausführung dieser und der anderen Routinen befinden, ist in Tabelle 9 angegeben.

OPEN und CLOSE

Nachdem wir alle Fileparameter und den Filenamen übergeben haben, können wir die OPEN-Routine mit JSR OPEN aufrufen. Schon ist das entsprechende File geöffnet. Zu beachten wäre folgendes: Es können im Computer niemals mehr als 10 Files gleichzeitig geöffnet sein!

Die CLOSE-Routine arbeitet analog zu OPEN, mit der Ausnahme, daß nur die Filenummer übergeben werden muß. Geräteadresse und Sekundäradresse sucht sich der C 64 aus einer Tabelle heraus, auf die wir später noch zu sprechen kommen:

```
LDA   #$01    ; Filenummer
JSR   CLOSE    in Akku
```

Der Filename wird beim Schließen überhaupt nicht mehr benötigt.

Auflistung aller verwendeten ROM-Routinen

Kürzel	Adresse	Kürzel	Adresse
		SECTLK	$FF96
		SECLST	$FF93
FILPAR	$FFBA	IECOUT	$FFA8
FILNAM	$FFBD	IECIN	$FFA5
OPEN	$FFC0	FILTAB	$F30F
CLOSE	$FFC3	FILSET	$F31F
LISTEN	$FFB1	LOAD	$FFD5
UNLIST	$FFAE	SAVE	$FFD8
TALK	$FFB4	BASOUT	$FFD2
UNTALK	$FFAB	CLALL	$FFE7

Tabelle 8. Die im Artikel erwähnten Betriebssystemroutinen der Floppy 1541

Wichtige Zero-Page-Adressen

Adresse	Bedeutung
$90	Status-Flag
$93	Flag für LOAD/VERIFY
$98	Anzahl der offenen Files
$99	Eingabegerät für BASIN
$9A	Ausgabegerät für BASOUT
$B7	Länge Filename
$B8	aktive Filenummer
$B9	Sekundäradresse
$BA	Geräteadresse
$BB/BC	Zeiger auf Filenamen

Tabelle 9. Dies sind Zero-Page-Adressen, unter denen die aktuellen Fileparameter gespeichert werden.

LISTEN und UNLISTEN, TALK und UNTALK

Nach dem Öffnen eines Files kann die Datenübertragung noch nicht beginnen. Sie müssen dem entsprechenden Gerät zuerst mitteilen, ob es senden oder empfangen soll.

Bestes Beispiel ist wieder der Kommandokanal. Über diesen kann das Floppy-Laufwerk sowohl Befehle empfangen, als auch Fehlermeldungen senden.

Um ein Gerät zum Empfangen zu veranlassen, verwenden wir die Routine LISTEN. Das hat nichts mit dem Basic-Befehl LIST zu tun, sondern kommt vom englischen Wort für »Hören«. Beim Aufruf von LISTEN ist das angesprochene Gerät auf Empfang und der Computer auf Senden eingestellt.

Wichtig ist, daß der Akku beim Aufruf die Geräteadresse enthält. Dies gilt für alle vier hier beschriebenen Routinen. Wenn Sie mit dem Senden der Daten fertig sind, sollten Sie ein UNLISTEN zum entsprechenden Gerät schicken, um dieses wieder freizugeben. Dies geschieht mit Hilfe der UNLIST-Routine. Analog verhält es sich mit den Routinen TALK und UNTALK. Sie veranlassen das angesprochene Gerät, Daten zu senden, beziehungsweise mit dem Senden aufzuhören und wieder in den Wartezustand zurückzukehren.

```
100 REM SCHREIBSCHUTZ SETZEN / LOESCHEN
101 REM DURCH AENDERUNG DES FORMAT-
102 REM KENNZEICHENS IN DER BAM !!!
103 REM FUNKTIONSWEISE SETZEN :
104 REM FORMATKENNZEICHEN WIRD AUF
105 REM BELIEBIGEN WERT AUSSER 'A'
106 REM GESETZT. AB SOFORT KOENNEN
107 REM KEINE SCHREIBVORGAENGE AUSSER
108 REM FORMATIEREN DURCHGEFUEHRT
109 REM WERDEN. ALSO VORSICHT !
110 REM FUNKTIONSWEISE FREIGEBEN :
111 REM DIE FLOPPY SPEICHERT DAS
112 REM FORMATKENNZEICHEN DER EINGEL.
113 REM DISKETTE IN DER SPEICHERSTELLE
114 REM $0101 ZWISCHEN. WIRD DIES
115 REM VOR DEM SCHREIBVORGANG AUF
116 REM 'A' ZURUECKGESETZT, LAESST
117 REM SICH DIE FLOPPY 'UEBERLISTEN'
118 REM DAS 'A' WIRD NUN IN DIE BAM
119 REM GESCHRIEBEN
120 :
130 PRINT"WOLLEN SIE DIE EINGEL. DISKETT
E"
140 INPUT"SCHUETZEN ODER FREIGEBEN";AA$
150 IF AA$="S"THEN 200
160 IF AA$="F"THEN 300
170 PRINT:RUN
180 :
190 :
200 OPEN 15,8,15,"I":OPEN8,8,8,"#"
210 PRINT#15,"U1 8 0 18 0"
220 PRINT#15,"B-P 8 2"
230 PRINT#8,"X";
240 PRINT#15,"U2 8 0 18 0"
250 PRINT#15,"I"
260 CLOSE8:CLOSE15:PRINT:RUN
270 :
280 :
290 :
300 OPEN 15,8,15,"I":OPEN8,8,8,"#"
310 PRINT#15,"U1 8 0 18 0"
320 PRINT#15,"B-P 8 2"
330 PRINT#15,"M-W"CHR$(1)CHR$(1)CHR$(1)C
HR$(65)
340 PRINT#8,"A";
350 PRINT#15,"U2 8 0 18 0"
360 PRINT#15,"I"
370 CLOSE8:CLOSE15:PRINT:RUN

READY.
```

Listing 6. Schützen Sie Ihre Diskette vor jedem Schreibzugriff

```
100 REM DIRECTORY-SORTER                    <180>
101 REM SORTIERT DIRECTORY ALPHABETISCH     <141>
102 REM BEI VIELEN EINTRAEGEN BITTE         <226>
103 REM ETWAS GEDULD (MAX. 5.MIN)           <219>
104 REM SORTIERT AUCH GESCRATCHTE FILES     <052>
105 REM MIT, STELLT SIE ABER NICHT          <087>
106 REM WIEDER HER ! SORTIERALGORITHMUS     <048>
107 REM KANN SICH IN EINEM SOLCHEN FALL     <121>
108 REM IN EINER ENDLOSSCHLEIFE VER-        <039>
109 REM HEDDERN. ABHILFE: NACH 3-4 MIN.     <009>
110 REM STOP-TASTE DRUECKEN, DANN           <143>
111 REM GOTO 210 EINGEBEN. SIND EINTR.      <019>
112 REM DANN NOCH NICHT VOLLKOMMEN SOR-     <227>
113 REM TIERT, NOCHMALS FUER EINIGE         <236>
114 REM MINUTEN LAUFEN LASSEN.              <208>
115 REM ACHTUNG !!! NUR ZUSAMMEN MIT        <190>
116 REM DEN UNTERPROGRAMMEN 1 & 2           <233>
117 REM ABLAUFFAEHIG !!!                    <182>
118 :                                       <176>
119 :                                       <177>
120 DIM DD$(144)                            <148>
130 MM=MM+1:GOSUB 1000                      <203>
140 IF DD$=NN$THEN MM=MM-1:GOTO 160         <248>
150 DD$(MM)=DD$:DD$="":GOTO 130             <190>
160 FOR GG=1 TO MM-1                        <172>
170 IF MID$(DD$(GG),4,16)<MID$(DD$(GG+1),4,
   16)THEN 190                              <054>
180 HH$=DD$(GG):DD$(GG)=DD$(GG+1):DD$(GG+1)=HH$
   :FF=1                                    <049>
190 NEXT GG                                 <206>
200 IF FF THEN FF=0:GOTO 160                <078>
210 II=MM                                   <176>
220 FOR MM=1 TO II:DD$=DD$(MM):GOSUB 2000
   :NEXT MM                                 <030>
230 END                                     <102>
```

Listing 7. Eine einfache Routine, um das Directory zu sortieren

```
100 REM BEISPIEL FUER MEMORY-EXECUTE
101 REM LOEST IN DER FLOPPY LANGSAMES
102 REM BLINKEN DER ROTEN LED (KENN-
103 REM ZEICHNET NORMALERWEISE HARD-
104 REM WARE-FEHLER) AUS.
105 REM KANN NUR DURCH AUSLOESEN EINES
106 REM RESETS ENTWEDER DURCH EIN/AUS-
107 REM SCHALTEN DER FLOPPY ODER DES
108 REM COMPUTERS BEENDET WERDEN.
109 REM EINSPRUNGSADRESSE : $EA6E
110 :
120 OPEN15,8,15
130 PRINT#15,"M-E"CHR$(110)CHR$(234)
140 CLOSE 15
150 END
READY.
```

Listing 8. Simulieren Sie einen Hardware-Fehler

SECTLK und SECLST

Die beiden Routinen SECTLK und SECLST sind ebenfalls sehr wichtig für die Datenübertragung. Denn obwohl wir beim OPEN-Befehl eine Sekundäradresse angeben, muß diese bei jeder weiteren Übertragung nochmals an das aktuelle Gerät gesendet werden.

Dies hat zwei Gründe: Einerseits können Sie ja mehrere Floppy-Kanäle gleichzeitig geöffnet halten. Damit die Floppystation nun weiß, für welchen Kanal der nächste Schwung von Daten bestimmt ist, beziehungsweise welcher Kanal senden soll, muß nach dem Aufruf von TALK SECTLK, beziehungsweise nach dem Aufruf von LISTEN SECLST durchgeführt werden. Außerdem merkt sich der Computer zwar die angegebene Sekundäradresse, sendet sie aber nicht.

Dies hat praktische Gründe, wie wir noch bei den LOAD/SAVE-Routinen sehen werden. SECTLK und SECLST benötigen die jeweilige Sekundäradresse +$60 im Akku. Diese kann, wie in unseren Beispielen, direkt geladen oder aber auch der entsprechenden Zero-Page-Adresse entnommen werden.

IECOUT und IECIN

Nachdem wir nun endlich alle Vorbereitungen getroffen haben, können wir munter Bytes von der Floppystation zum Computer und umgekehrt übertragen. Dies ist mit den ROM-Routinen denkbar einfach. IECOUT überträgt das im Akku befindliche Byte an das aktuelle Gerät; IECIN empfängt eines und legt es im Akku ab.

Bei aller Sorgfalt - Fehler können immer auftreten, so auch beim Busbetrieb. Um einen in einer Busroutine aufgetretenen Fehler zu signalisieren, verwendet das Betriebssystem das Carry-Flag. Generell gilt: Ist das Carry-Flag gesetzt, so ist etwas nicht in Ordnung, und wir sollten das Status-Byte überprüfen. Dieses Status-Byte steht in der

Speicherstelle $90. Immer wenn es ungleich Null ist, liegt irgendein Sonderfall vor. Jedes Bit des Status-Bytes hat eine andere Funktion; Tabelle 10 zeigt diese Belegung. Ist zum Beispiel das Bit 7 gesetzt, so ist das angesprochene Gerät entweder nicht vorhanden oder abgeschaltet. In Basic bekämen wir in einem solchen Fall die Meldung »DEVICE NOT PRESENT ERROR«. Interessant ist für uns noch das Bit 6. Ist es gesetzt, so bedeutet das, daß das letzte Byte der angeforderten Informationen übertragen wurde. Dies können wir uns auch in Basic zunutze machen, um beispielsweise die Fehlermeldung der Floppystation auszulesen:

```
10 OPEN 1, 8, 15
20 GET#  1, A$: PRINTA$;: IF ST<>64 THEN 20
30 CLOSE 1
```

Wie Sie an diesem Beispiel sehen, ist der Inhalt der Speicherstelle $90 in der Variablen ST enthalten. Vor jeder neuen Datenübertragung sollten Sie darauf achten, daß das Status-Byte gelöscht wird, da sonst irrtümlich Fehler festgestellt werden könnten. Zur Verdeutlichung dienen die Listings 9 und 10, die jedoch nur Anhaltspunkte geben sollen. Sie sind weder perfekt noch eintippfertig und sollten auf den jeweiligen Bedarf abgestimmt werden.

Bearbeiten mehrerer Files

Sie werden festgestellt haben, daß wir bisher immer nur mit einem einzigen File gearbeitet haben. Was aber, wenn Sie gleichzeitig zwei Files offenhalten müssen, zum Beispiel, um einen Block von Diskette zu lesen. Sie erinnern sich ja, daß wir dazu sowohl den Kommandokanal als auch einen Übertragungskanal benötigen. Wir könnten zwar jeweils, wenn wir den Kanal wechseln wollen, mit CLOSE den alten schließen und mit OPEN den neuen öffnen, aber es geht auch einfacher.

Voraussetzung ist, daß alle benötigten Files schon geöffnet sind. Dann kann mit Hilfe einer, schon erwähnten, Filetabelle zwischen – bis zu 10 – Files beliebig umgeschaltet

Das Status-Flag

Bit	Bedeutung wenn gesetzt
1	Fehler (Zeitüberschreitung) bei IEC-Eingabe
2	Fehler (Zeitüberschreitung) bei IEC-Ausgabe
3-5	nur für Kassettenbetrieb
6	Übertragung ist beendet
7	Gerät meldet sich nicht

Tabelle 10. Für uns wichtige Bits im Status-Flag

werden. Diesen Zweck erfüllen die Routinen FILTAB und FILSET.

FILTAB benötigt im Akku die Nummer des Files, auf das Sie umschalten wollen. Die Routine sucht dann in der Filetabelle nach den entsprechenden anderen Parametern. Tritt hier ein Fehler auf, weil das File noch gar nicht geöffnet wurde, so wird das Zero-Flag gelöscht und es kann mit BNE auf einen Fehler überprüft werden.

FILSET schreibt dann die gefundenen Parameter in die entsprechenden Zero-Page-Adressen. Die komplette Routine zum Umschalten auf das File x lautet also:

```
LDA    #$ xx     ; Nummer des Files
JSR    FILTAB    ; Durchsuchen der Tabelle
BNE    ERROR     ; Fehler?
JSR    FILSET    ; Parameter setzen
```

Die ERROR-Routine müssen Sie natürlich noch selbst schreiben. Danach ist das angewählte File zum aktuellen File geworden. Alle LISTEN, TALK und so weiter beziehen sich jetzt auf dieses neue File.

In den Zero-Page-Adressen (Tabelle 9) stehen nun die für dieses File aktuellen Parameter, da sie aus der großen Filetabelle automatisch übertragen werden. Eine Ausnahme bildet hier der Filename, da er nur beim Öffnen des Files benötigt wird.

```
(In $FA/$FB muß die Adresse, in $FC die Länge des
Befehlsstrings stehen.)

    LDA #$01     ; Filenummer
    LDX #$08     ; Gerätenummer
    LDY #$6F     ; Sekundäradresse
    JSR FILPAR   ; setzen.
    LDA #$00     ; Länge Filename=0
    JSR FILNAM   ; da kein Filename.
    JSR OPEN     ; File öffnen
    LDA #$08     ; Geräteadresse
    JSR LISTEN   ; auf Empfangen
    LDA #$6F     ; Sekundäradresse
    JSR SECLST   ; senden.
    LDY #$00     ; Zähler auf Null
X   LDA ($FA),Y  ; Befehlsbyte laden
    JSR IECOUT   ; und übertragen
    INY          ; Zähler erhöhen
    CPY #$FC     ; Befehlslänge
    BNE X        ; noch ein Byte?
    LDA #$08     ; Geräteadresse
    JSR UNLIST   ; Sendung beenden
    LDA #$01     ; Filenummer
    JSR CLOSE    ; Schliessen
    RTS
```

Listing 9. So können Befehlsstrings an die Floppy gesendet werden

```
Prinzip des Lesens des Fehlerkanals mit Ausgabe
auf dem Bildschirm

    LDA #$00     ; Zurücksetzen des
    STA  $90     ; Status-Flags
    LDA #$01     ; Filenummer
    LDX #$08     ; Geräteadresse
    LDY #$6F     ; Sekundäradresse
    JSR FILPAR   ; setzen.
    LDA #$00     ; Länge Filename=0
    JSR FILNAM   ; setzen.
    JSR OPEN     ; File öffnen
    LDA #$08     ; Geräteadresse auf
    JSR TALK     ; Senden schalten
    LDA #$6F     ; Sekundäradresse
    JSR SECTLK   ; übertragen
X   JSR IECIN    ; Byte empfangen
    JSR BASOUT   ; und ausgeben
    BIT  $90     ; Bit 6 Status=0?
    BVC X        ; dann noch ein Byte
    LDA #$08     ; Geräteadresse
    JSR UNTALK   ; Sendung beenden
    LDA #$01     ; Filenummer
    JSR CLOSE    ; und schliessen
    RTS
```

Listing 10. So läßt sich der Fehlerkanal auslesen und anzeigen

Die große File-Tabelle in der Zero-Page befindet sich übrigens an den Speicherstellen $0259 bis $0276.

Denken Sie immer daran, vor einem erneuten Umschalten UNLIST oder UNTALK aufzurufen. CLOSE braucht dagegen erst aufgerufen zu werden, wenn die Bearbeitung eines Files völlig abgeschlossen ist.

LOAD und SAVE

Prinzipiell könnten Sie mit dem bisher Erwähnten auch schon Programme laden und speichern, allerdings nur sehr mühselig. Da unser Computer das aber schon von selbst beherrscht, geben wir ihm gern diese Arbeit ab.

Betrachten wir zunächst die LOAD-Routine. Auch hier muß wieder eine Vielzahl an Parametern übergeben werden. Mit FILPAR werden Gerätenummer und Sekundäradresse gesetzt. Eine Filenummer braucht nicht gesetzt zu werden. Für die Sekundäradresse gilt folgendes:

Ist sie gleich Null, so wird das Programm an eine, von Ihnen festgelegte, Speicherstelle geladen. Ist sie gleich Eins, so wird das Programm an die Speicherstelle geladen, an der es bei SAVE stand. Der erste Modus wird vom Betriebssystem ausgenutzt, um Programme ab $0800 zu laden, wenn beim LOAD-Befehl keine Sekundäradresse angegeben wird. Prinzipiell kann aber an jede beliebige Adresse geladen werden! Der Filename wird, wie gewohnt, mit FILNAM gesetzt. Vor dem Aufruf der LOAD-Routine treten zwei, uns neue, Parameter hinzu, die wie folgt übergeben werden:

LOAD/VERIFY-Flag (Akku)
Ladeadresse LO (X-Register)
Ladeadresse HI (Y-Register)

Steht beim Aufruf der Routine im Akku 0, so wird geladen. Steht dort hingegen eine 1, so wird ein VERIFY durchgeführt.

Die Startadresse in den X/Y-Registern wird nur beachtet, wenn die Sekundäradresse gleich Null ist. Alles übrige erledigt die LOAD-Routine, und Sie brauchen nur noch deren Ende abzuwarten. Zur Sekundäradresse wäre noch folgendes zu bemerken:

Egal, was Sie für eine Adresse angeben, zur Floppystation wird immer nur 0 gesendet. Wie Sie schon wissen, ist diese Sekundäradresse laufwerksintern für den LOAD-Befehl reserviert und darf nicht ohne weiteres bei OPEN-Befehlen verwendet werden. Nach Beendigung der LOAD-Routine wird im X und Y-Register die Endadresse des Programms übergeben.

Die SAVE-Routine hat eine etwas kompliziertere Parameterübergabe. FILPAR braucht nur mit der Gerätenummer im X-Register aufgerufen zu werden, da weder Sekundäradresse noch Filenummer benötigt werden. Das Setzen des Filenamens erfolgt normal über FILNAM.

Übergeben werden müssen nun noch Anfangsadresse und Endadresse+1 des zu speichernden Bereichs. Die Anfangsadressen müssen Sie irgendwo in der Zero-Page in der Reihenfolge LO/HI ablegen. Empfehlenswert wären die Adressen $FB/FC, da diese nicht vom Betriebssystem oder Basic benutzt werden. Im Akku muß dann die Adresse des LO-Byte übergeben werden; wenn Sie die Adresse also unter $FB/FC speichern, muß im Akku $FB stehen.

Die Endadresse übergeben Sie wie folgt:
LO-Byte im X- und HI-Byte im Y-Register. Es muß immer 1 zur Endadresse addiert werden, da sonst das letzte Byte des Programms nicht gespeichert wird. Danach kann die Routine SAVE aufgerufen werden. Wieder haben wir für Sie zur Verdeutlichung zwei Listings: Listing 11 zeigt, wie man ein Programm an eine beliebige Adresse lädt; Listing 12, wie man einen beliebigen Bereich auf Diskette speichert.

```
Prinzip des Ladens von Programmen.

LDX #$08    ;  Geräteadresse
LDY #$00    ;  Sekundäradresse für
               relativ laden
JSR FILPAR  ;  und setzen.
LDX # (Filename LO-Byte)
LDY # (Filename HI-Byte)
LDA # (Filename Länge)
JSR FILNAM
LDA #$00    ;  LOAD-Flag
LDX # (Startadresse LO-Byte)
LDY # (Startadresse HI-Byte)
JSR LOAD
RTS
```

Listing 11. Das Laden von Programmen an beliebige Adressen funktioniert so

```
Prinzip des Speicherns von
Bereichen.

LDX #$08    ;  Geräteadresse
JSR FILPAR  ;  setzen
LDX # (Filename LO-Byte)
LDY # (Filename HI-Byte)
LDA # (Filename Länge)
JSR FILNAM  ;  setzen
LDX # (Startadresse LO-Byte)
LDY # (Startadresse HI-Byte)
STX $FB     ;  zwischenspeichern
STY $FC
LDA #$FB    ;  Pointer auf Startadr.
LDX # (Endadresse +1 LO-Byte)
LDY # (Endadresse +1 HI-Byte)
JSR SAVE
RTS
```

Listing 12. Und so funktioniert das Speichern

Erwähnenswert ist noch die Routine CLALL, die alle Files im Computer schließt; die Kanäle in der 1541 bleiben davon jedoch unberührt. Hier müssen Sie also sorgfältig mit CLOSE arbeiten, da Sie sonst Daten verlieren können.

Spooling? Was ist das?

Nachdem wir Sie nun mit Theorie überschwemmt haben, sollen Sie sogleich in den Genuß Ihrer neuen Kenntnisse kommen. Haben Sie schon einmal etwas von Spooling gehört? Nein? Macht nichts, wir werden uns mit dieser Technik nämlich jetzt auseinandersetzen, und Sie werden dabei die Vorzüge dieser Möglichkeit genießen lernen.

Unter dem Begriff Spooling verbirgt sich eigentlich eine ganz einfache Technik, die jedoch enorme Vorteile besitzt: Es handelt sich um das Drucken direkt von Diskette. Haben Sie nicht auch schon öfters versucht, ein meterlanges Listing auf Papier zu bringen und den Drucker dabei mit wütenden Blicken zu größerer Eile aufgefordert, weil Sie nämlich unter Zeitdruck standen und sich bei der Arbeit keine Verzögerung erlauben konnten? Dann ist Spooling genau das Richtige für Sie. Bei dieser Methode wird ein Listing, das ausgedruckt werden soll, auf Diskette gebracht. Danach starten Sie ein Spooling-Programm und siehe da; der Drucker beginnt, Ihr Listing auf Papier zu bringen, und der Computer meldet sich betriebsbereit mit READY.

Dies ist kein Wunder, sondern die Eigenschaft des seriellen Bus Ihres Computers. Sie haben vorhin gelernt, wie

man den Bus des Computers in Maschinensprache bedient. Dabei fielen auch Worte wie TALK, LISTEN, SENDEN und EMPFANGEN. Der Trick des Spooling ist nun der: Mit Hilfe des CMD-Befehls in Basic können Sie ein Listing auf Diskette »umleiten« und zwar geschieht dies ähnlich wie beim Drucker: Sie öffnen ein File und schicken mit dem CMD-Kommando alle weiteren Bildschirmausgaben auf den Bus. Nur ist jetzt nicht der Drucker der Adressat, sondern die Floppystation. Hier ein Beispiel:
Sie haben ein Listing im Speicher und wollen dieses auf Diskette ablegen, sein Name soll »TEST« sein:

```
OPEN 1, 8, 2, "TEST,U,W"
CMD 1
LIST
```

Nach dieser Befehlsfolge wird Ihr Listing als USR-File auf Diskette geschrieben. Wie wäre es nun, wenn die Floppystation ein TALK-Kommando erhalten würde, das sie veranlaßt, das eben geschriebene File auf den Bus zu bringen? Der »Hörer« ist aber jetzt nicht, wie üblich, der Computer, sondern der Drucker, den wir zuvor mit einem LISTEN dazu aufgefordert haben. Die Folge wäre das, was Sie sich jetzt schon denken können:

Die Floppystation schickt das gesamte Listing über den Bus, und der Drucker, der ja auf Empfang programmiert ist, bekommt dieses Listing und druckt es aus. Der Computer hat mit der ganzen Sache nichts zu tun, da er sich nach Senden der Kommandos »zurückgezogen« hat und bleibt demzufolge frei für weitere Arbeit.

Drucken ohne Umwege

Der Zugriff auf den Bus ist dem Computer natürlich für die Zeit der Übertragung verwehrt, aber Sie können währenddessen intern weiterarbeiten. Ist die Übertragung beendet, so sind beide Peripheriegeräte noch auf Sendung beziehungsweise Empfang und müssen erst »zur Ruhe gebracht« werden, bevor sie wieder ansprechbar sind. Aber auch das erledigt ein kleines Programm für uns. Sehen Sie sich jetzt einmal Listing 13 an. Es enthält ein Spooling-Programm, das mit

```
SYS 828, "filename"
```

aufgerufen wird. Danach meldet sich der Computer mit
SPOOLING filename
READY
und der Drucker beginnt zu arbeiten. Ist der Druckvorgang beendet, so tippen Sie noch einmal

```
SYS828
```

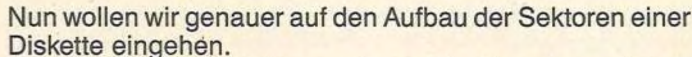
ohne Filenamen, und die Leuchtdiode an der Floppy erlischt. Es erscheint die Meldung
END OF SPOOLING
READY
Dieses Programm ist, im Gegensatz zu unseren anderen Listings, zum sofortigen Eintippen mit einem Assembler oder besser noch, einem Monitor, gedacht.

Wir mußten jedoch leider feststellen, daß das Spooling nicht mit allen Druckern funktioniert. Bitte probieren Sie das von Fall zu Fall aus.

Wie Sie aus diesem Beispiel sehen, kann es von großem Nutzen sein, wenn Sie das Prinzip des seriellen Bus verstehen und dessen »Verkehrsregeln« kennen, da viele Programme nur deshalb mit geringem Aufwand große Effekte und Nutzen erzielen. Ein weiteres Beispiel in dieser Reihe dürfte wohl Hypra-Load sein, das Sie in Ausgabe 10/1984 des 64'er-Magazins fanden. Dieses Programm nutzt aber noch einige weitere Tricks der Maschinenspracheprogrammierung, die wir später besprechen wollen. Im folgenden wollen wir nämlich in die direkte Programmierung der Floppystation einsteigen, das heißt, das Speichern von Maschinenprogrammen in ihren Pufferspeicher und das Ausführen derselben mittels spezieller Befehle.

```
., 033C 20 79 00 JSR $0079
., 033F F0 43    BEQ $0384
., 0341 20 E7 FF JSR $FFE7
., 0344 A0 00    LDY #$00'
., 0346 B9 A5 03 LDA $03A5,Y
., 0349 F0 06    BEQ $0351
., 034B 20 D2 FF JSR $FFD2
., 034E C8       INY
., 034F D0 F5    BNE $0346
., 0351 20 54 E2 JSR $E254
., 0354 20 C1 F5 JSR $F5C1
., 0357 A6 B7    LDX $B7
., 0359 F0 66    BEQ $03C1
., 035B A9 01    LDA #$01'
., 035D A2 08    LDX #$08'
., 035F A0 02    LDY #$02'
., 0361 20 BA FF JSR $FFBA
., 0364 20 C0 FF JSR $FFC0
., 0367 A9 04    LDA #$04'
., 0369 20 B1 FF JSR $FFB1
., 036C 20 BE ED JSR $EDBE
., 036F A2 01    LDX #$01'
., 0371 20 C6 FF JSR $FFC6
., 0374 20 BE ED JSR $EDBE
., 0377 20 85 EE JSR $EE85
., 037A 20 97 EE JSR $EE97
., 037D A9 00    LDA #$00'
., 037F 85 99    STA $99
., 0381 85 98    STA $98
., 0383 60       RTS
., 0384 A9 01    LDA #$01'
., 0386 85 98    STA $98
., 0388 20 AE FF JSR $FFAE
., 038B 20 AB FF JSR $FFAB
., 038E A9 01    LDA #$01'
., 0390 20 C3 FF JSR $FFC3
., 0393 A0 00    LDY #$00'
., 0395 B9 AF 03 LDA $03AF,Y
., 0398 F0 06    BEQ $03A0
., 039A 20 D2 FF JSR $FFD2
., 039D C8       INY
., 039E D0 F5    BNE $0395
., 03A0 A9 00    LDA #$00'
., 03A2 4C 74 A4 JMP $A474
.
.
.
.'03A5 53 50 4F 4F 4C 49 4E 47
.'03AD 20 00 45 4E 44 20 4F 46
.'03B5 20 53 50 4F 4F 4C 49 4E
.'03BD 47 8D 00 00 4C 08 AF 00
.
.
.
., 03C1 4C 08 AF JMP $AF08
.
.
.
READY.
```

Listing 13. Mit diesem Programm können Sie ein Floppy-Drucker-Spooling durchführen. Näheres im Text.



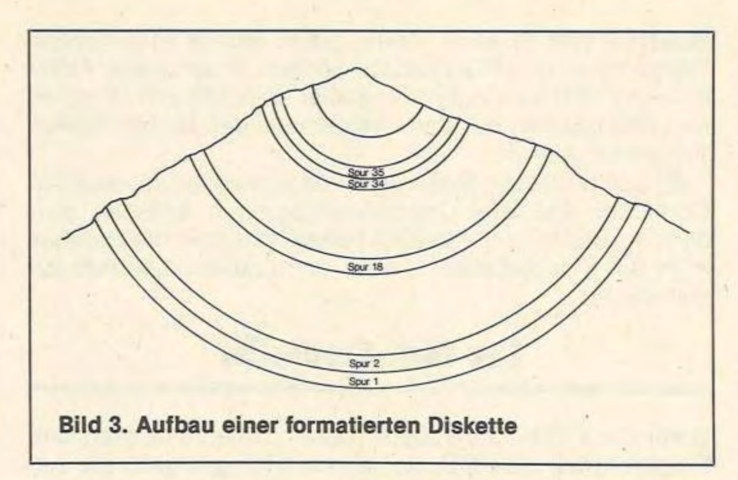


Bild 3. Aufbau einer formatierten Diskette

Nun wollen wir uns mit dem Aufzeichnungsformat der Diskette beschäftigen: Für einen einwandfreien Betrieb der Floppystation ist es unumgänglich, daß sich Markierungen auf der Diskette befinden. Diese Markierungen braucht das Laufwerk, um bestimmte Daten schnell finden zu können. Hierfür gibt es prinzipiell zwei Möglichkeiten: die Hardsektorierung und die Softsektorierung.

Hardsektorierte Disketten erkennt man daran, daß diese eine ganze Anzahl von Indexlöchern besitzen. Damit sind die kleinen Löcher nahe am Innenrand der Magnetscheibe gemeint. Mit einer Fotozelle können nun diese Löcher abgetastet werden, um die jeweilige Position der Diskette festzustellen. Dieses Verfahren hat den Vorteil, daß die Diskettenkapazität voll ausgenutzt werden kann. Es können so bis zu 5 MByte Daten auf eine 5¼-Zoll-Diskette geschrieben werden. Allerdings erfordert diese Methode einen enormen Hardware-Aufwand, der den Preis in die Höhe schnellen läßt. Für preiswerte Laufwerke (wie die 1541) geht man daher einen anderen Weg: die Softsektorierung. Hier besitzt die Diskette nur ein Indexloch zur Erkennung des Trackbeginns. Bei der 1541 ist sogar noch nicht einmal dieses erforderlich. Die notwendigen Markierungen werden beim Formatierungsvorgang softwaremäßig auf die Diskette gebracht, wobei natürlich wertvoller Speicherplatz verlorengeht. Softsektorierte Disketten im 5¼-Zoll-Format verfügen daher über zur Zeit maximal 1,2 MByte Speicherkapazität.

Uns soll also im weiteren die Softsektorierung beschäftigen, wobei in Bild 3 eine Diskette schematisch dargestellt ist, nachdem sie auf der 1541 formatiert wurde. Sie ist in 35 konzentrische Spuren, nachfolgend Tracks genannt, aufgeteilt. Jeder dieser Tracks enthält wiederum eine bestimmte Anzahl von Sektoren, die von außen nach innen abnimmt. Diese Tatsachen sind Ihnen aber schon bekannt.

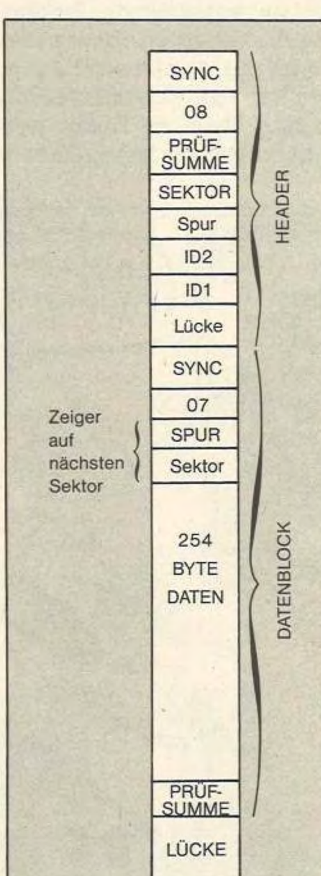


Bild 4. Aufbau eines Sektors, einer mit der Floppy 1541 formatierten Diskette

Nun wollen wir genauer auf den Aufbau der Sektoren einer Diskette eingehen.

Jeder Sektor besteht aus einem Blockheader und dem dazugehörigen Datenblock; eine schematische Darstellung zeigt Bild 4. Angeführt werden die Sektoren einer Diskette von den schon erwähnten Markierungen, die der Orientierung dienen. Diese Marken bezeichnet man als Synchron (SYNC)-Markierungen, sie bestehen aus mehreren $FF auf der Diskette. Erkennt der Schreib-/Lesekopf der Floppy also eine solche Marke, dann »weiß« die Floppystation, daß entweder ein Blockheader oder ein Datenblock nachfolgt. Nun müssen wir nur noch diese beiden voneinander unterscheiden können.

Hierzu dient das nächste Kennzeichen auf Diskette. Es folgt direkt nach der SYNC-Markierung und meldet dem Diskcontroller (DC), ob ein Blockheader oder ein Datenblock vorliegt. Hat das Kennzeichen den Wert $08, so handelt es sich um einen Blockheader; findet der Kopf hingegen den Wert $07, so handelt es sich um den Beginn eines Datenblocks.

Wir nehmen jetzt einmal an, der DC hätte das Kennzeichen $08 entdeckt; es handelt sich also um den Header eines Datenblocks. Dann folgt als nächstes Byte die Prüfsumme über den Header, die zur Kontrolle auf Lesefehler dient. Die Reihenfolge der Header-Bytes, wie sie im Commodore-Handbuch angegeben ist, stimmt nicht mit der Aufzeichnung auf Diskette überein.

Arbeitsweise des DOS

Die nächsten zwei Byte stellen Sektor- und Track-Nummer dieses Sektors dar. Anhand dieser Werte kann der DC bei Track-Wechsel sehr schnell die Position des Schreib-/Lesekopfes ausfindig machen.

Das 5. und 6. Byte des Blockheaders gibt jeweils einen Teil der ID der Diskette an, und zwar folgen zuerst das zweite und dann das erste Zeichen der ID, die beim Formatieren festgelegt wurden. Mit diesen Angaben ist die Behandlung des Headers bereits abgeschlossen. Es folgen jetzt noch ein paar Bytes, die eine Lücke darstellen.

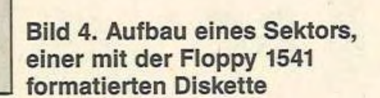
Mit der nächsten SYNC-Markierung wird der Beginn des eigentlichen Datenblocks eingeleitet. Nach der SYNC-Marke folgt das Datenblockkennzeichen $07. Die nächsten zwei Byte sind uns bestens bekannt. Sie können mit jedem Diskmonitor angesehen werden und geben Track- und Sektornummer des nächsten Blocks im File an. Man bezeichnet sie deshalb als Linker oder Link-Adressen (engl.: to link = verbinden).

Nun erst folgen die eigentlichen Daten auf Diskette, die in jedem Block 254 Byte ausmachen. Hinter diesen Daten-Bytes steht die Prüfsumme des Datenblocks, die wiederum zum Erkennen von eventuellen Lesefehlern dient. Werden solche Fehler festgestellt, so versucht die Floppystation noch mehrere Male, den Block doch zu lesen. Erst wenn viele Versuche kein befriedigendes Ergebnis bringen, steigt sie mit einer Fehlermeldung aus.

Nach der Prüfsumme des Datenblocks folgt wieder eine Lücke auf der Diskette, bevor die SYNC-Markierung des nächsten Blockheaders kommt. Wenn wir uns diesen Aufbau eines Sektors betrachten, wird klar, warum die Speicherkapazität bei softsektorierten Disketten gegenüber hardsektorierten Disketten deutlich abnimmt.

Jetzt werden Sie vielleicht auch die Beschreibung der Fehlermeldungen im Floppy-Handbuch verstehen, die dort sehr genau und richtig erläutert werden.

Das Verständnis des Diskettenaufbaus ist für die weitere Behandlung des DOS unerläßlich, da wir nur so die Funktionsweise begreifen lernen.



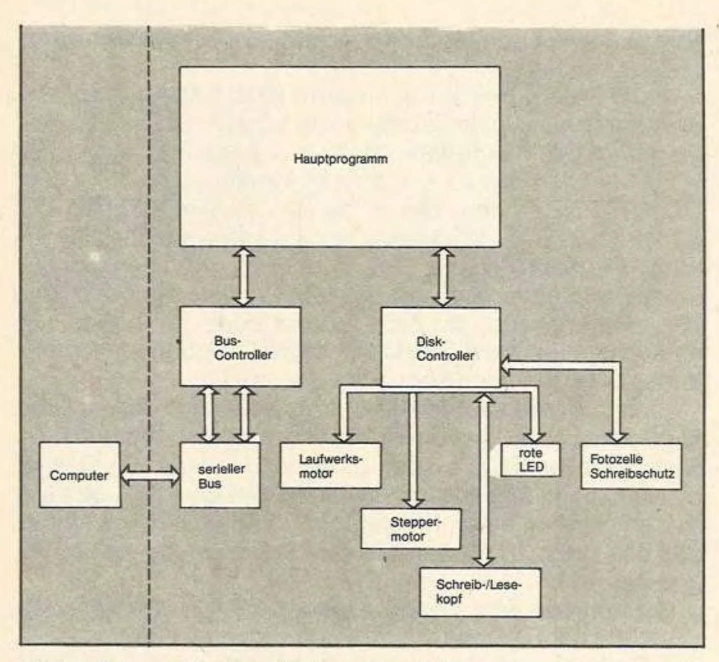


Bild 5. Schematischer Aufbau der 1541

Jetzt wollen wir uns aber einmal mit der grundlegenden Arbeitsweise des Floppy-Betriebssystems (DOS) befassen, das um einiges komplizierter ist, als das des Computers.

Wenn wir das Laufwerk einschalten, passiert zunächst das gleiche wie im Computer. Die RESET-Leitung geht auf Low und der Mikroprozessor, hier ein 6502, holt sich seine Systemstartadresse. Danach läuft das RESET-Programm an, wobei die Floppystation einen Selbsttest durchführt. Erkennen können Sie dies daran, daß für kurze Zeit der Motor anläuft und die rote LED leuchtet. Wurde kein Defekt registriert, so erlischt die Leuchtdiode wieder, und der Motor geht aus. Jetzt wird der RAM-Bereich der Floppy initialisiert und alle wichtigen Zeiger werden hergestellt. Danach ist die 1541 betriebsbereit.

Von jetzt an laufen quasi drei Programme gleichzeitig ab:
- das Hauptprogramm läuft in einer Schleife, die nur bei der Ausführung von Befehlen verlassen wird;
- das Disk-Controller-Programm wird über den IRQ gesteuert und durch den Timer des Disk-Controllers (DC) alle 10 ms aufgerufen;
- die Routinen des Bus-Controllers (BC) schließlich werden nur im Bedarfsfall aufgerufen, nämlich, wenn die ATN-Leitung des seriellen Bus auf Low geht.

Wir wollen uns die Funktion dieser Routinen nun einmal etwas genauer betrachten.

Das Hauptprogramm

Das Hauptprogramm hängt, wie schon gesagt, in einer Warteschleife, bis ein Befehl vom Computer kommt. Dieser aktiviert zuerst die Busroutinen, die die gesendeten Bytes dann entgegennehmen und speichern. Jetzt bekommt das Hauptprogramm, das übrigens den Zustand der beiden Steuer-Routinen (DC und BC) ständig überwacht, die Meldung, daß ein Befehl anliegt. Es verzweigt nun zur Befehlsauswertung, ähnlich dem Basic-Interpreter, und führt gegebenenfalls einen Befehl aus, sofern kein Syntaxfehler entdeckt wurde. In diesem Fall würde sonst eine Fehlermeldung generiert, die dann vom Computer ausgelesen werden kann.

Ist ein Befehl korrekt ausgeführt worden, so werden die Befehlsparameter wieder gelöscht, und das Hauptprogramm kehrt in die Warteschleife zurück.

Das Disk-Controller-Programm

Der Disk-Controller enthält den Baustein VIA 6522, durch den er mit dem Mikroprozessor in Kontakt steht. Dieser Baustein enthält auch Timer, die in einem eingestellten Rhythmus einen IRQ auslösen können. Einer dieser Timer ist in der 1541 so eingestellt, daß er ungefähr alle 10 ms einen IRQ auslöst, der dann seinerseits das Diskcontroller-Programm aufruft.

Es soll an dieser Stelle der Unterschied zwischen Disk-Controller und Disk-Controller-Programm erläutert werden: Als Disk-Controller (DC) bezeichnet man die Hardware in der Floppystation, die für den Laufwerksbetrieb zuständig ist.

Der Disk-Controller

Unter dem Disk-Controller-Programm versteht man den Programmteil im DOS, der durch IRQ geregelt, die Ansteuerung des DC übernimmt.

Eine vollständige Trennung dieser beiden Begriffe ist jedoch weder notwendig noch zweckmäßig, so daß wir mit dem Ausdruck »DC« immer die Gesamtheit von Hard- und Software beschreiben wollen. Nun aber wieder zu den Aufgaben des DC.

Auch dieses Programm hat eine Art Wartezustand, solange kein Befehl vom Computer anliegt. Wird nämlich das Hauptprogramm über den Bus aktiviert, so wertet dieses die Befehle aus und gibt sie an den DC weiter, der dann seinerseits dafür sorgt, daß das Laufwerk aktiviert wird. Er steuert den Laufwerks- und den Stepper(Schreib-/Lesekopf)-Motor und bedient die Daten, die vom und zum Schreib-/Lesekopf gehen. Die gesamten Vorgänge am Laufwerk werden also interruptgesteuert vorgenommen.

Die Busroutinen

Die Routinen des Bus-Controllers (BC) werden ebenfalls über die IRQ-Leitung gesteuert. Auch der BC enthält einen VIA 6522-Baustein. Hier wird der Aufruf der Routinen allerdings nicht über den Timer organisiert, sondern, wie schon erwähnt, über die ATN-Leitung des seriellen Bus. Zieht der Computer also diese Leitung auf Low, so wird in der Floppystation (und in allen anderen Peripheriegeräten ebenso) ein IRQ ausgelöst. Dann erfolgt die Abfrage, ob dieser IRQ vom Timer des DC kam. Ist dies nicht der Fall, so wird die BC-Routine aufgerufen, die dann den weiteren Busbetrieb übernimmt. Sollte die Floppystation gerade einen Befehl



$0000	Jobspeicher für Puffer 0
$0001	Jobspeicher für Puffer 1
$0002	Jobspeicher für Puffer 2
$0003	Jobspeicher für Puffer 3
$0004	Jobspeicher für Puffer 4
$0005	Jobspeicher für Puffer 5 (im RAM nicht vorhanden)
$0006/7	Spur- und Sektornummer für Befehl in Puffer 0
$0008/9	Spur- und Sektornummer für Befehl in Puffer 1
$000A/B	Spur- und Sektornummer für Befehl in Puffer 2
$000C/D	Spur- und Sektornummer für Befehl in Puffer 3
$000E/F	Spur- und Sektornummer für Befehl in Puffer 4
$0010/1	Spur- und Sektornummer für Befehl in Puffer 5
$0012/3	ID der Diskette im ASCII-Code: die beiden Zeichen der aktuellen ID werden bei jedem Blocksuchbefehl gelesen und hier aktualisiert abgespeichert. Auch das Initialisierkommando benutzt diesen Befehl und bringt die ID dadurch auf den neuesten Stand.
$0016-$001A	Hier sind die Bytes für den aktuellen Blockheader gespeichert, und zwar sind dies: $0016 erstes Zeichen der ID $0017 zweites Zeichen der ID $0018 Spurnummer des Blocks $0019 Sektornummer des Blocks $001A Prüfsumme über den Blockheader Auf der Diskette stehen diese Werte in der umgekehrten Reihenfolge!
$001C	Flag für Änderung beim Schreibschutz der Diskette
$002E/F	Zwischenspeicher für aktuelle Zeiger
$0030/1	Zeiger in aktuellen Puffer
$0032/3	Zeiger auf aktuellen Blockheader beim Schreiben
$0038	Kennzeichen ($07) für Beginn eines Datenblocks
$0039	Kennzeichen ($08) für Beginn eines Blockheaders
$003A	Zwischenspeicher für Prüfsummen
$003D	aktuelle Laufwerksnummer; bei der VC 1541 immer 0
$003E	gerade arbeitendes Laufwerk ($FF = kein Laufwerk)
$003F	Puffernummer des eben ausgeführten Befehls (0-5)
$0043	zählt die Anzahl der Sektoren bei der Formatierung
$0044	Zwischenspeicher beim Arbeiten
$0045	Zwischenspeicher für aktuellen Befehlscode
$0047	enthält aktuelles Kennzeichen für Beginn eines Datenblocks. wird nur bei RESET einmal auf $07 gesetzt und kann vom Benutzer verändert werden, wobei das Hi-Nybble des Wertes immer auf 0 ($0-) stehen sollte, um Leseprobleme des DC zu vermeiden. Wird versucht, einen Datenblock mit einer anderen, als der hier gespeicherten, Nummer zu lesen, so erfolgt der Fehlercode $04 des DC und die Floppy sendet Fehlermeldung Nummer 22 zum Bus.
$0049	Zwischenspeicher für den Stackpointer
$004A	Zähler für Kopftransport; Zahlen bis 127 bewegen den Kopf nach außen; Zahlen von 128 bis 255 bewegen ihn nach innen (höhere Spurnummer).
$0051	aktuelle Spurnummer bei der Formatierung; steht auf $FF, wenn keine Formatierung erfolgt.
$0065/6	Zeiger auf die NMI-Routine; wird bei einem RESET gestellt.
$0067	Flag zum Anzeigen eines NMI
$0068	Flag zum Ermöglichen (0) oder Sperren (1) der automatischen Initialisierung einer Diskette, falls ein ID Type Mismatch Error erkannt wurde
$0069	Abstand der Sektoren bei der Zuteilung; erhält bei einem RESET den Wert 10.
$006A	Anzahl der Leseversuche eines Sektors; steht nach RESET auf 5.
$006B/C	Zeiger auf Sprungtabelle der USER-Befehle; steht normalerweise auf $FFF6 nach einem RESET.
$006D/E	Zeiger auf den Beginn der 'Bit Map'; steht auf $0400 und wird beim Initialisieren gesetzt.
$006F	Zwischenspeicher; steht nach RESET auf $6F
$0070	Zwischenspeicher
$0071	Zwischenspeicher
$0072	Zwischenspeicher; steht nach RESET auf $FF
$0073	Zwischenspeicher
$0074	Zwischenspeicher
$0075/6	Indirekter Zeiger auf $0100; wird bei RESET gestellt
$0077	Gerätenummer + $20 für das LISTEN-Kommando
$0078	Gerätenummer + $40 für das TALK-Kommando
$0079	Flag für LISTEN (1/0)
$007A	Flag für TALK (1/0)
$007B	Flag für Adressierung
$007C	Flag für ATN-Signal vom seriellen Bus
$007D	Flag für Prozessor im ATN-Modus
$007F	Aktuelle Laufwerksnummer; hier immer 0
$0080	Aktuelle Spurnummer; enthält $00 nach Ausführung
$0081	Aktuelle Sektornummer; enthält $00 nach Ausführung
$0082	Aktuelle Kanalnummer
$0083	Aktuelle Sekundäradresse
$0084	übliche Sekundäradresse
$0085	Aktuelles Datenbyte
$0086	Speicher für Zwischenergebnisse
$0087	Speicher für Zwischenergebnisse
$0088	Speicher für Zwischenergebnisse
$0089	Speicher für Zwischenergebnisse
$008A	Speicher für Zwischenergebnisse
$008B-$008E	Speicher für Ergebnisse bei Berechnungen
$008F-$0093	Akkumulator für Berechnungen
$0094/5	Zeiger auf Directory-Puffer; enthält $05/02
$0096	Kommando vom IEEE-Bus; hier unbenutzt
$0098	Bitzähler für seriellen Bus
$0099/A	Buffer-Pointer für Puffer 0; steht auf $0300
$009B/C	Buffer-Pointer für Puffer 1; steht auf $0400
$009D/E	Buffer-Pointer für Puffer 2; steht auf $0500
$009F/0	Buffer-Pointer für Puffer 3; steht auf $0600
$00A1/2	Buffer-Pointer für Puffer 4; steht auf $0700 Alle diese Pointer werden durch den B-P-Befehl verändert!
$00A3/4	Zeiger auf nächstes Zeichen im INPUT-BUFFER ($0200)
$00A5/6	Zeiger auf nächstes Zeichen im ERROR-BUFFER ($02D6)
$00A7-$00AD	Tabelle; enthält für jeden aktiven Puffer die entsprechende Kanalnummer. Kanalnummer = $FF, wenn Puffer unbenutzt.
$00AE-$00B4	Tabelle; enthält für jeden aktiven Puffer die entsprechende Kanalnummer. Kanalnummer = $FF, wenn Puffer unbenutzt.
$00B5-$00BA	Tabelle der Lo-Bytes der Recordnummern für jeden Puffer
$00BB-$00C0	Tabelle der Hi-Bytes der Recordnummern für jeden Puffer
$00C1-$00C6	Tabelle der nächsten zu bearbeitenden Recordnummern für jeden Puffer
$00C7-$00CC	Tabelle der Recordlängen für jeden Puffer
$00CD-$00D2	Tabelle der Side-Sektoren für jeden Puffer
$00E2-$00E6	Standardwerte für Laufwerk; hier alle 0
$00E7-$00EB	Tabelle der Filetypen
$00EC-$00F1	Kanal Filetyp
$00F2-$00F7	Kanalstatus
$00F8	Zwischenspeicher für EOI
$00F9	Aktuelle Puffernummer für Befehlscode
$0101	Formatkennzeichen von Spur 18 Sektor 0
$0104-$0145	Bereich des Hardware-Stack; nicht benutzbar
$0200-$0229	INPUT-BUFFER; hier werden alle Befehlsstrings vom Computer zwischengespeichert und nach Syntaxprüfung aus geführt
$022A	Codenummer des auszuführenden Befehls
$022B-	Kanaltabelle; diese Tabelle enthält für jede mögliche
$023E-$0243	Aktuelles Datenbyte für jeden Kanal; Belegung der Adressen wie bei der Kanalstatustabelle ($022B)
$0244-$0249	Tabelle der Zeiger auf das letzte aktuelle Zeichen in jedem, für den Kanal zuständigen, Pufferspeicher
$024A	Gerade behandelter Filetyp
$024B	Länge des Befehlsstrings
$024C	Zwischenspeicher für Sekundäradresse
$024D	Zwischenspeicher für Befehlscode
$024E	Arbeitsspeicher beim Suchen des nächsten Sektors
$024F/0	Pufferbelegungsspeicher; 1 = Puffer belegt
$0253	Flag für Directory-Eintrag gefunden
$0254	Flag für $-Befehl zum Listen des Directory
$0255	Flag für Befehlsausführung (<>$00, wenn Befehl anliegt)
$0257	Nummer des letzten benutzten Puffers
$0258	Recordlänge
$0259	Side-Sector Spur
$025A	Side-Sector Sektor
$025B-$025F	Tabelle; enthält den letzten Befehlscode der Puffer
$0260-$0265	Sektornummern der Directoryeinträge in den Puffern
$0266-$026B	Zeiger auf die Directoryeinträge in den Puffern
$026D	Flag für LED Blinken bei Fehler
$026E	Nummer des letzten aktiven Laufwerks
$026F	Nummer des letzten bearbeitenden Sektors
$0270	aktueller Schreibkanal
$0271	aktueller Lesekanal
$0274	Länge des Befehlsstrings im INPUT-BUFFER
$027A-$027F	Tabelle der Zeiger auf die Filenamen
$0280-$0284	Spurnummern der Files für den aktuellen Puffer
$0285-$0289	Sektornummern der Files für den aktuellen Puffer
$028A	Joker (*) Flag
$028E	Standardwert für die Nummer des Laufwerks
$028F	Flag für Fileeintrag im Directory gefunden
$0290	Sektornummer des aktuellen Directory Sektors
$0291	Sektornummer des ersten Directoryeintrags
$0292	Zeiger auf ersten gültigen Directoryeintrag
$0293	Zeigt letzten Block an; enthält dann 0
$0294	Aktueller Pufferzeiger
$0295	Zähler für Fileeinträge
$0297	Betriebsart des aktuellen Files (Lesen/Schreiben)
$029D/E	Spurnummer der BAM
$02A1-$02B0	Zwischenspeicher für BAM Eintragungen
$02B1-$02D4	Puffer für Directory
$02D5-$02F8	ERROR-BUFFER; enthält auszugebende Fehlermeldung
$02FA	Lo-Byte der Anzahl der freien Blocks auf Diskette
$02FC	Hi-Byte der Anzahl der freien Blocks auf Diskette
$0300-$03FF	Puffer 0
$0400-$04FF	Puffer 1
$0500-$05FF	Puffer 2
$0600-$06FF	Puffer 3
$0700-$07FF	Puffer 4 (enthält normalerweise die BAM)
$0800-$FFFF	Nicht mit RAM belegt

Tabelle 11. Die wichtigsten Zero-Page-Adressen der Floppy

bearbeiten, während schon ein neuer vom Computer gesendet wird, so wartet der BC so lange mit der Annahme, bis das Laufwerk wieder in den Bereitschaftszustand zurückgekehrt ist.

Wie Sie sehen, stellt das DOS eine ziemlich komplizierte Einheit dar, deren Schema in Bild 5 zu sehen ist.

Wollen wir also in dieses System einsteigen, um dort eigene Programme ausführen zu lassen, so ist es natürlich unerläßlich, daß wir die »Spielregeln« dieses Prozessorsystems genau kennen, da es sonst leicht zu kleinen Katastrophen kommen kann.

Disketten werden auf ihrer Rückseite beschrieben!

Zu Ihrer weiteren Arbeit mit der 1541/70/71 noch ein paar Tips: Wenn Sie vorhaben, Programme im Floppy-RAM ablaufen zu lassen, sollten Sie Ihre Floppystation öffnen und ohne Deckel betreiben. So können Sie genau beobachten, wie der Kopf positioniert wird und was bei Lesefehlern geschieht. Sie werden unter anderem auch entdecken, daß Disketten nicht etwa auf der Seite beschrieben werden, auf der sich das Etikett befindet, sondern auf der Rückseite. Dies ist um so bemerkenswerter, als man eine Diskette immer nur auf der Vorderseite schonend behandelt, die ja eigentlich nicht benutzt wird. Auch wir mußten die Erfahrung machen, daß wir Disketten lange Zeit mit der wertvollen Seite auf Tische gelegt haben, stets darauf achtend, daß ja kein Staubkorn auf die gehütete Vorderseite kam.

Das Betreiben ohne Deckel hat auch den Vorteil besserer Wärmeableitung. Die Mechanik wird es Ihnen danken.

Nachdem Sie Ihre 1541 also auf »Arbeitsbetrieb« getrimmt haben, wollen wir gleich einmal mit kleinen Programmen beginnen. In Tabelle 11 sehen Sie eine Aufstellung einiger wichtiger Zero-Page-Adressen, die uns im weiteren Verlauf noch beschäftigen werden. Für ein DOS-Listing ist in unserer Serie natürlich kein Platz vorhanden; auch können wir nur mit kleinen Beispielen versuchen, Ihnen die Programmierung der 1541 nahezubringen. Für diejenigen unter Ihnen, die jedoch vorhaben, tiefer in die Floppy-Programmierung einzusteigen, seien an dieser Stelle zwei Bücher angesprochen, die von Markt & Technik herausgegeben werden und die 1541 und die 1570/71 bis ins kleinste Detail behandeln. Sie enthalten außerdem sehr gut dokumentierte DOS-Listings und gehen weit über das in diesem Kurs besprochene hinaus.

Programmieren der Floppy

So, jetzt soll es aber endlich losgehen. Wir wollen unser erstes Programm schreiben und in der Floppystation ablaufen lassen.

Es handelt sich um Listing 14. Dieses »Miniprogramm« schreiben wir in den Puffer 0 der Floppystation, das heißt ab Adresse $0300. Das Basic-Programm haben wir der Kürze halber gleich an den Assemblercode angehängt. Wenn Sie das Programm starten, wird das Bit abgefragt, das beim DC für den Zustand der Schreibschutzplakette verantwortlich ist. Sie werden vielleicht wissen, daß das Laufwerk die Schreibschutzkerbe bei den Disketten mit Hilfe einer Lichtschranke abfragt. Ist die Lichtschranke unterbrochen, das heißt es liegt eine Diskette mit Schreibschutzaufkleber im Laufwerk, dann steht das entsprechende Bit auf 0. Durchsichtiger Tesafilm kann deshalb nicht als Schreibschutz verwendet werden.

Unser Programm schiebt nun einfach das Bit der Lichtschranke an die Stelle des Bits für die rote LED und speichert diesen Wert wieder. Starten Sie einmal unser kleines Programm, dann werden Sie feststellen, daß die Leuchtdiode am Laufwerk erlischt, wenn die Lichtschranke unterbrochen wird. Holen Sie die Diskette dagegen aus dem Laufwerk oder legen Sie eine Diskette ohne Schreibschutzplakette ein, so beginnt die rote LED zu leuchten.

```
0 GOTO 10                                          <234>
1 , 0300 AD 00 1C LDA $1C00                        <018>
2 , 0303 29 10    AND #$10                         <023>
3 , 0305 4A       LSR                              <093>
4 , 0306 8D 00 1C STA $1C00                        <041>
5 , 0309 4C 00 03 JMP $0300                        <005>
6 :                                                <064>
10 OPEN 1,8,15                                     <208>
20 FOR X=0 TO 11:READ A                            <215>
30 PRINT#1,"M-W"CHR$(X)CHR$(3)CHR$(1)CHR$(A)
   :NEXT                                           <065>
40 PRINT#1,"M-E"CHR$(0)CHR$(3)                     <179>
50 DATA 173,0.28,41,16,74,141,0,28,76,0,3          <005>
```

Listing 14. Unser erstes Floppy-Maschinenprogramm

```
100 REM ERZEUGT &-FILE, DAS LISTING 2              <220>
110 REM (LED-TEST) ENTSPRICHT.                     <194>
120 :                                              <178>
130 DATA 0,7,12,173,0,28,41,16,74,141              <093>
140 DATA 0,28,76,0,7,93                            <197>
150 OPEN 1,8,2,"&,U,W"                             <194>
160 FOR X=1 TO 16:READ A                           <105>
170 PRINT#1,CHR$(A);:NEXT X                        <071>
180 CLOSE 1                                        <133>
```

Listing 15. So macht man Listing 14 zu einem »&-File«

```
100 REM AUTO-'&'-MAKER                             <106>
110 REM --------------                             <115>
120 REM                                            <007>
130 REM  03.11.84.  BORIS SCHNEIDER                <224>
140 :                                              <198>
150 :                                              <208>
160 REM INITIALISIERUNG                            <159>
170 INPUT"STARTADRESSE DES &-FILES";SA             <121>
180 INPUT"NAME DES &-FILES";NA$                    <046>
190 IF LEN(NA$)>15 THEN 180NA$                     <026>
200 OPEN 1,8,2,"&"+NA$+",U,W"                      <063>
210 DIM X(256)                                     <158>
220 PRINT"BITTE GEBEN SIE JETZT IHRE DATEN EIN"
                    <116>
230 PRINT"ABSCHLUSS MIT -1!"                       <212>
240 :                                              <042>
250 REM DATENEINGABE UND TEST AUF                  <227>
260 REM UEBERLAUF                                  <047>
270 Y=1                                            <075>
280 INPUT X(Y)                                     <160>
290 IF X(Y)<0 THEN Y=Y-1:GOTO 350                  <213>
300 PR=PR+X(Y):IF PR>255 THEN PR=PR-255            <103>
305 Y=Y+1:IF Y>254 THEN 350                        <026>
310 GOTO 280                                       <090>
320 :                                              <123>
330 REM ABSPEICHERN DER VORHANDENEN                <017>
340 REM DATEN IN DAS USR-FILE                      <006>
350 SH=INT(SA/256)                                 <144>
360 SL=SA-256*SH                                   <221>
370 PR=PR+SH+SL+Y                                  <250>
380 PRINT#1,CHR$(SL);CHR$(SH);                     <082>
390 PRINT#1,CHR$(Y);                               <040>
400 FOR I=1 TO Y                                   <059>
410 PRINT#1,CHR$(X(I));                            <213>
420 NEXT                                           <039>
430 PR=PR-(255*INT(PR/256))                        <219>
440 PRINT#1,CHR$(PR);                              <163>
450 IF X(Y+1)<0 THEN GOTO 470                      <217>
460 SA=SA+Y:PR=0:GOTO 270                          <196>
470 CLOSE 1                                        <168>
```

Listing 16. Komfortabel »&-Files« erzeugen

Mit diesem Programm können Sie also testen, ob von Ihnen selbst angefertigte Schreibschutzkerben in der Diskettenhülle an der richtigen Stelle liegen, um eine Diskette eventuell doppelseitig benutzen zu können.

Da unser Programm aus einer Endlosschleife besteht, können Sie die Floppystation nur durch einen Reset wieder in einen ansprechbaren Zustand versetzen.

DISKCONTROLLER (DC)

VIA 6522, $1800, PORT B

Bit #	Bedeutung
0	DATA IN
1	DATA OUT
2	CLOCK IN
3	CLOCK OUT
4	ATN OUT
5	GERÄTENUMMER
6	
7	ATN IN (CB 2)

BUSCONTROLLER (BC)

VIA 6522, $1C00, PORT B

Bit #	Bedeutung
0	Steppermotor - Spule 1
1	Steppermotor - Spule 2
2	Laufwerksmotor
3	LED am Laufwerk (rot)
4	Schreibschutzkennung
5	Bitsynchronisation für DC
6	bei den vier Spurbereichen
7	SYNC-Signal

Tabelle 12. Belegung der beiden Control-Ports der 1541

Das Programm hat aber einen Schönheitsfehler; es beeinflußt nämlich nicht nur die beiden LED-Bits in Speicherstelle $1C00, sondern löscht bei jedem Durchgang auch alle anderen Bits dieses Registers, deren Belegung Sie Tabelle 12 entnehmen können. Für unsere Testzwecke ist diese »Pfuscherei« jedoch unwesentlich.

Der »&«-Befehl

Nach diesem aufregenden Beispiel wollen wir Sie nun mit einem Befehl bekanntmachen, den Sie sehr wahrscheinlich noch nicht kennen. Er nennt sich »&« und wird unverständlicherweise in keiner Anleitung gut beschrieben. Der &-Befehl entspricht in gewisser Weise einem BLOCK-EXECUTE-Befehl; auch hier wird ein Programm von Diskette geladen und sofort ausgeführt.

Der Unterschied besteht nur darin, daß mit dem &-Befehl nicht nur ein Block, sondern ein ganzes File, das im Directory verzeichnet ist, geladen und im Puffer als Programm ausgeführt wird.

Außerdem müssen die Files, die mit dem Befehl »&« gestartet werden sollen, speziell gekennzeichnet sein. Sie enthalten als erstes Zeichen im Filenamen das Zeichen »&«. Soll also zum Beispiel ein File mit dem Namen »Test« als Autostartprogramm in der 1541 ausgeführt werden, so geben Sie diesem File den Namen »&Test« und starten Sie es danach mit:

```
OPEN1,8,15," &TEST"
```

Haben Sie nur ein einziges Autostartfile auf Diskette, so können Sie es auch nur mit »&« abspeichern und ebenso mit

`OPEN1,8,15,"&"` starten.

Leider erwartet die Floppystation von Autostart-Files eine spezielle Syntax, die in Tabelle 13 zu sehen ist.

Als Listing 15 haben wir noch einmal unser LED-Testprogramm; nur wird diese Routine durch das Basic-Programm als &-File auf Diskette geschrieben, und kann danach durch den schon erwähnten Befehl direkt von Diskette in den Pufferspeicher geschrieben und dort gestartet werden.

Zu Tabelle 13 noch einige Anmerkungen:

Zuerst muß die Startadresse des Programms im Pufferspeicher der 1541 in das File geschrieben werden. Danach folgt die Anzahl der Bytes im Programm. Jetzt werden die Programmbytes gespeichert, und schließlich folgt noch eine Prüfsumme, die sich wie folgt errechnet:

Komplizierte Prüfsummenberechnung

Es werden alle Bytes des Programms addiert und zum Ergebnis noch die zwei Byte der Startadresse und die Anzahl der Bytes im Programm hinzugezählt. Dieses Ergebnis ist als Integerzahl zu verstehen und besteht also aus einem niederwertigen (LO) und einem höherwertigen (HI) Byte. Das niederwertige Byte ist die Prüfsumme, zu der noch der Übertrag im höherwertigen Byte addiert werden muß. Diese Berechnung klingt kompliziert; ist es aber nicht. Die allgemeine Formel hier noch einmal:

HB=INT(SUMME/256)
LB=SUMME-HBx256

dabei bedeuten:

HB - das höherwertige Byte
LB - das niederwertige Byte
SUMME - die Gesamtsumme der Programm-Bytes

Achtung: Die Übertragsberechnung muß nach jedem neu dazugezählten Wert erfolgen, da das Endergebnis kleiner als 256 sein muß! Wie Sie sehen, ist das Anlegen eines &-Files nicht ganz einfach. Bisher wurde diese Fileart fast nur von Profis zum Programmschutz angewandt, da sie, wie schon erwähnt, nahezu unbekannt war.

Zu erwähnen wären noch zwei seltsame Fehlermeldungen der 1541:

»OVERFLOW IN RECORD« erscheint, wenn die Anzahl der tatsächlichen Bytes mit der Angabe nicht übereinstimmt.

»RECORD NOT PRESENT« erscheint, wenn die Prüfsumme nicht stimmt.

Da wir stets darum bemüht sind, Ihnen die Arbeit mit der Floppystation so angenehm wie möglich zu machen, haben wir unserem Artikel noch Listing 16 beigefügt. Es handelt sich hier um ein Programm, das es Ihnen gestattet, auf einfachste Weise &-Files zu erstellen. Diese können sogar länger als 256 Byte sein, da das Programm dann automatisch eine Prüfsumme und die Anschlußadresse einfügt. Ununterbrochene &-Files, die länger als 256 Zeichen sind, kann es ja nicht geben, da die Anzahl der Programm-Bytes im File nur in einem Byte gespeichert wird.

Die fortgeschrittenen Programmierer unter Ihnen werden sicher schon mit Ungeduld auf den Beginn des folgen-

Byte	Bedeutung
1-2	Startadresse in der 1541 im HI/LO-Format
3	Anzahl der folgenden Programmbytes
4-N	Programm
N+1	Prüfsumme
N+2	Hier kann bei längeren Programmen ein weiterer Teil eingefügt werden. Format: wieder bei Byte 1 beginnend.

Tabelle 13. Aufbau eines &-Files. In dieser Tabelle sind die Linker- beziehungsweise Endekennzeichen, die in den ersten beiden Byte eines Datenblocks stehen, nicht enthalten, da sie beim Öffnen und Beschreiben eines &-Files automatisch gesetzt werden.

den Abschnitts gewartet haben. Jetzt wird unser Kurs seinem Titel nämlich endlich voll gerecht werden, und wir wollen einmal sehen, was sich so alles mit einer Diskette anstellen läßt. Selbstverständlich sollen dabei Errors (Diskettenfehler) und »Killertracks« auch nicht zu kurz kommen.

Damit wir uns aber wieder an wichtige Tatsachen erinnern, noch einmal eine kurze Zusammenfassung einiger wichtiger Einzelheiten.

Wie ausführlich beschrieben, besteht ein Sektor auf Diskette aus zwei Teilen, nämlich dem Header und dem eigentlichen Datenblock. Beide Teile des Sektors werden auf Diskette durch eine SYNC-Markierung angekündigt, der dann das Kennzeichen (ob Header- oder Datenblock) zur Identifikation folgt.

Wichtige Unterprogramme des DOS:

$FD9E	– Rücksprung in die Jobschleife
$F556	– Sync-Signal auf Diskette abwarten
$FE00	– PCR auf Lesen umschalten
$FE0E	– Track mit $55 vollschreiben
$FDA3	– Track mit SYNC vollschreiben
$F510	– Blockheader lesen:
	+ Diskette muß initialisiert sein
	+ $32/33 muß die Adresse der Track- und Sektornummer enthalten (L/H); zum Beispiel $00/03, wenn die Nummern in $0300/0301 abgespeichert sind
	+ Rückkehr nur bei fehlerfreier Durchführung des Lesens
$F527	– Blockheader lesen:
	+ Diskette muß initialisiert sein
	+ zuvor muß $12 nach $16 und $13 nach $17 gebracht werden
	+ Track- und Sektornummer in $18 und $19
	+ Rückkehr nur bei fehlerfreier Durchführung des Lesens
$F50A	– Datenblockanfang suchen:
	+ Parameter siehe $F510

Tabelle 14. Einige Unterprogramme des DOS, die in den Beispielprogrammen verwendet werden

Der Blockheader enthält Track- und Sektornummer des Blocks, die beiden Bytes der Diskettenidentifikation (ID) und schließlich noch eine Prüfsumme, die dem Diskcontroller (DC) mitteilt, ob alle Daten einwandfrei gelesen wurden. Wurde der Blockheader richtig eingelesen, so wartet der DC auf den nachfolgenden Datenblock, der die Zeiger auf den nächsten Block im File, die Daten-Bytes und schließlich ebenfalls eine Prüfsumme enthält.

Zwischen Blockheader und Datenblock und zwischen Datenblock und Header des darauffolgenden Sektors befindet sich jeweils eine Lücke, die dem DC Zeit zum Umschalten seiner Modi (Lesen und Schreiben) läßt und außerdem für eine symmetrische Verteilung der Sektoren auf Diskette sorgt.

So, und jetzt genug der Wiederholung. Wir werden uns auf ein paar grundsätzliche Programmbeispiele stürzen, die Sie später in eigene Anwendungen einbauen können.

Wie wir schon wissen, werden alle Schreib-/Lesevorgänge des Disk-Controllers interruptgesteuert vorgenommen. Es ist also zum direkten Eingriff auf Diskette notwendig, daß wir uns die Regeln der Interruptsteuerung genau einprägen, da uns die Floppystation bei unseren Experimenten sonst mit Sicherheit »abstürzt«.

Da wir in unserem Kurs verständlicherweise kein DOS-Listing abdrucken können, wurden die wichtigsten Adressen, die wir benötigen, in Tabelle 14 zusammengefaßt und mit einer kurzen Erläuterung versehen, damit Sie sich mit der Anwendung der DOS-Routinen vertraut machen können.

Ein weiteres »Werkzeug« ist die RAM-Belegung der wichtigsten Speicherstellen.

Um den Einstieg zu finden, fangen wir gleich einmal mit der Übergabe der Kommandos an den DC an. Wie bewerkstelligt es das Hauptprogramm, die unterschiedlichsten Befehle wie Lesen, Schreiben, Suchen, Kopf bewegen, Laufwerksmotor an, Formatieren und so weiter an den Disk-Controller zu übergeben?

Um eine Antwort auf diese Frage zu finden, betrachten Sie sich bitte die Tabelle 15. Sie enthält eine Aufstellung aller Jobcodes der Floppy 1541. Mit Jobcodes sind hierbei die Kommandos gemeint, die dafür sorgen, daß ein bestimmter Job zur Ausführung kommt.

Nehmen Sie jetzt einmal die Belegung der Zero-Page zur Hand. Wenn Sie sich die Speicherstellen $0000 bis $0005 betrachten, so merken Sie schon am Namen, daß diese Adressen etwas mit unserer Sache zu tun haben. Es handelt sich hierbei um die Jobspeicher, die die Aufgabe haben, für den Dialog zwischen Hauptprogramm und DC zu sorgen.

Wurde eben etwas von Dialog (nicht etwa Monolog) gesagt? Genau! Die Jobspeicher dienen nicht nur der Übergabe der Kommandos vom Hauptprogramm an den Disk-Controller; sie enthalten nach der Ausführung des Jobs auch die Rückmeldung des DC, an der das Hauptprogramm erkennen kann, ob der Job erfolgreich, das heißt fehlerlos durchgeführt worden ist.

Die Rückmeldungen des Disk-Controllers sind komplett in Tabelle 16 aufgeführt. Wenn Sie sich einmal die Bitmuster der Jobcodes und der Rückmeldungen ansehen und beide Typen miteinander vergleichen, so werden Sie sehr schnell einen Unterschied feststellen, der von entscheidender Bedeutung ist:

Die Befehls-Codes sind ausschließlich negative Werte, das heißt Werte, die größer als $80 (128) sind. Das Kennzeichen solcher Zahlen ist das gesetzte Bit 7 im Byte, das deshalb auch als »negative bit« bezeichnet wird und bei jeder Befehlsausführung in Maschinensprache direkt in das Prozessorstatusregister übernommen wird (N-Flag).

Die Rückmeldungen sind fast ausschließlich Zahlen, die kleiner als $0F (16) sind (bis auf eine Ausnahme). Diese Größe spielt zwar nicht direkt eine Rolle; das Wichtigste ist jedoch, daß bei diesen Werten keiner größer als $7F (127) ist. Zu der Begründung für diese Einteilung werden wir im folgenden noch kommen.

Wie Sie aus der Belegung der Zero-Page ersehen, existiert für jeden Puffer der Floppystation ein eigener Jobspeicher. Das ermöglicht einen sehr dynamischen Einsatz der Floppystation, der es zum Beispiel erlaubt, mit mehreren Puffern gleichzeitig zu arbeiten.

Eine wichtige Regel sollten Sie sich gleich einprägen, damit später keine Pannen passieren: Wenn Sie einen Jobcode an den DC übergeben, sollten Sie darauf achten, daß der DC für die Ausführung dieses Jobs meistens einen Puffer benötigt. Den Puffer, der dabei zum Beispiel beschrieben wird, wählen Sie durch die Übergabe des Jobcodes in der entsprechenden Speicherstelle aus.

Achtung: Verwenden Sie dabei niemals den Puffer, in dem Sie Ihr Programm abgelegt haben, da dieses sonst un-

Jobcodes des DOS:

$80	– Lesen eines Blocks in einen Puffer
$90	– Schreiben eines Blocks aus einem Puffer
$A0	– Verify eines Sektors mit einem Pufferinhalt
$B0	– Testen eines Sektors auf Vorhandensein
$C0	– 'Bump' des Tonkopfes
$D0	– Maschinenprogramm im Puffer ausführen
$E0	– Programm im Puffer ausführen, nachdem das Laufwerk hochgefahren ist

Tabelle 15. Zeigt alle Jobcodes mit deren Aufgaben

ter Umständen gelöscht wird und sich die Diskettenstation auf »mysteriöse« Weise verabschiedet.

Haben Sie also beispielsweise ein Programm ab $0300 (Puffer 0) abgelegt, so sollten Sie sich davor hüten, die Zero-Page-Adresse $0000 als Jobspeicher zu benutzen.

Auch als Zwischenspeicher sind die Adressen $0000 bis $0005 nicht unbedingt zu empfehlen, da es sonst zu einer kleinen Katastrophe kommen kann.

Die Kommandos an den Disk-Controller

Haben Sie sich die Speicherbelegung der Floppystation schon etwas genauer betrachtet, so werden Ihnen auch die Speicherstellen $0006 bis $0011 nicht entgangen sein.

Wie wir wissen, gibt es verschiedene Jobcodes, die bestimmte Aktionen hervorrufen (die ausführliche Erläuterung der Jobcodes folgt gleich). Nun ist es aber in der Regel notwendig, einem Befehl auch ein paar Parameter mitzugeben, die dann in entsprechender Weise abgearbeitet werden.

In unserem Fall sind das sicherlich die Track- und Sektornummern, auf die sich unser jeweiliger Befehl beziehen soll. Wie Sie aus der Tabelle 15 nämlich ersehen können, existiert zum Beispiel ein Jobcode, der das Lesen eines Blocks veranlaßt. Hier ist es natürlich nötig, die Blockparameter mit anzugeben.

Wollen Sie also ein Kommando $80 an den DC für Puffer 1 übergeben, so schreiben Sie zunächst in die Speicherstelle $0008 die Track-Nummer und in Speicherstelle $0009 die Sektornummer des Blocks, der in Puffer 1 gelesen werden soll. Anschließend erhält die Speicherstelle $0001 den Jobcode und auf geht's... Das klingt alles recht einfach. Stimmt, recht viel komplizierter wird es auch nicht mehr.

Unser einziges Problem besteht jetzt nur noch in der Tatsache, daß der DC für die Ausführung der Befehle eine gewisse Zeit benötigt, die je nach Kommando mehrere Interruptaufrufe erforderlich macht. Woher wissen wir also jetzt, wann ein Block vollständig in den Puffer gelesen ist und wir dessen Inhalt übernehmen können?

Die Lösung dieses Problems liegt in der unterschiedlichen Wertigkeit der Befehls-Bytes und der Rückmeldungen des DC, die ich vorhin schon angesprochen habe. Sie können sich noch erinnern: Alle Jobcodes bestehen aus Werten größer als $80 und alle Rückmeldungen aus Werten kleiner als $80.

Da der DC aber nach jedem Job seine Rückmeldung in der gleichen Speicherstelle hinterläßt, in die wir vorher das Kommando geschrieben hatten, ist es uns nun ein leichtes, diese Speicherstelle zu überprüfen und das Ende des Jobs anhand der Rückmeldung abzufragen. Anhand der noch folgenden Beispiele wird diese Technik gründlich erläutert.

Jetzt wollen wir uns aber mit den eigentlichen Jobcodes und deren Aufgaben beschäftigen.

1) Lesen eines Sektors in einen Puffer:

Wenn wir einen Sektor in einen Puffer lesen wollen, so stellen wir fest, daß diese Aktion auf der Ebene der Jobschleife fast genauso einfach ist, wie von Basic aus mit dem »B-R«- beziehungsweise »U1«-Befehl. Zum Lesen eines Sektors geben Sie dessen Track- und Sektornummer in den entsprechenden Speicherstellen für den gewünschten Puffer an. Anschließend senden Sie den Code $80 an den DC, und das Laufwerk startet sofort und liest den Sektor ein.

Diese Befehlsübergabe können Sie sogar von Basic aus, mit den MEMORY- und BLOCK-Befehlen, realisieren und dann den Pufferinhalt auslesen, um sich zu überzeugen, daß der Block auch wirklich eingelesen wurde.

Achtung: Die Diskette muß beim Arbeiten in der Jobschleife von Hand initialisiert werden, da wir uns auf dieser unteren Programmierebene im Rücken der automatischen Initialisierung befinden, die hier deshalb nicht mehr von alleine erfolgt. Merken Sie, daß der Inhalt im Puffer nicht mit

Rückmeldungen der Jobschleife:

$01	- Fehlerfreie Durchführung (00, OK)
$02	- Blockheader wurde nicht gefunden (20, Read Error)
$03	- SYNC-Markierung nicht gefunden (21, Read Error)
$04	- Datenblock wurde nicht gefunden (22, Read Error)
$05	- Datenprüfsumme ist falsch (23, Read Error)
$07	- Fehler nach einem Verify (25, Write Error)
$08	- Diskette ist schreibgeschützt (26, Write Peotect on)
$09	- Prüfsumme im Header falsch (27, Read Error)
$0A	- Datenblock auf Diskette zu lang (28, Write Error)
$0B	- Falsche ID im Blockheader (29, Disk ID Mismatch)
$0F	- Keine Diskette im Laufwerk (74, Drive not ready)
$10	- Fehler bei Dekodierung (24, Read Error)

Tabelle 16. Zeigt alle Rückmeldungen des DC, wobei in Klammern die zugehörige Fehlermeldung steht

dem auf der Diskette übereinstimmt, so kann das mit großer Wahrscheinlichkeit an der fehlenden Initialisierung liegen; doch auch dazu später noch mehr.

Jetzt wollen wir die Jobcodes anhand kleiner Beispiele genauer kennenlernen; dabei wollen wir uns auch gleichzeitig mit den Rückmeldungen des DC vertraut machen, anhand derer sich Fehler in der Ausführung des Jobs erkennen lassen.

Wir werden jetzt den Jobcode für Lesen des Blocks 18,1 in Puffer 0 übergeben und uns dann die Rückmeldung und den Inhalt des Blocks ansehen. Mit dem POKE-Befehl im Programm schreiben wir den Inhalt des Puffers direkt in den Bildschirmspeicher, was für unsere Kontrolle langen soll:

```
1 OPEN 1, 8, 15, "I"
2 PRINT#1, " M-W"  CHR$(6) CHR$(0) CHR$(2)
  CHR$(18) CHR$(1)
3 PRINT#1, " M-W"  CHR$(0) CHR$(0) CHR$(1)
  CHR$(128)
4 FORX=0TO2000: NEXT X
5 PRINT#1, " M-R"  CHR$(0) CHR$(0) CHR$(1)
6 GET#1, A$: PRINT ASC(A$+CHR$(0))
7 FORX=0TO255
8 PRINT#1, " M-R"  CHR$(X) CHR$(3) CHR$(1)
9 GET#1, A$: POKE1024+X,ASC(A$+CHR$(0))
10 NEXT X
11 CLOSE 1
```

Dieses kleine Programm initialisiert die Diskette im Laufwerk. Anschließend werden Track und Sektor (18,1) übergeben und schließlich der Jobcode in Adresse $0000 geschrieben, der dafür sorgt, daß unser Block in Puffer 0 geladen wird. Nach einer kleinen Warteschleife, in der das Laufwerk Zeit zur Befehlsausführung hat, wird der Jobspeicher wieder ausgelesen. Anhand von Tabelle 15 können Sie erkennen, daß der Job ordnungsgemäß ausgeführt wurde, wenn Sie als Rückmeldung eine »1« bekommen.

Auf dem Bildschirm erscheint der Inhalt des Puffers, wobei unter anderem auch Teile des Directory der Diskette zum Vorschein kommen sollten.

2) Schreiben eines Blocks auf Diskette:

Analog zum Lesen eines Blocks erfolgt das Schreiben. Hier übergeben Sie die gleichen Parameter; nur muß sich der zu schreibende Block schon im Puffer der Floppystation befinden. Durch die Auswahl des Job-Speichers können Sie jeden x-beliebigen Puffer des Laufwerks (0 bis 4) in jeden Block der Diskette schreiben.

3) Verifizieren eines Blocks von Diskette:
Dieser Vorgang erfolgt in der Floppystation bei einem SAVE normalerweise automatisch. Aus diesem Grund dauert das Speichern eines Programms auch um einiges länger als das Wiedereinladen in den Computer. Mit Hilfe des entsprechenden Jobcodes ($A0) können wir ein Verify aber nach Belieben starten, um den Inhalt in einem Pufferspeicher mit einem Block auf Diskette zu vergleichen.

```
0500          ; LOESCHEN VON TRACK 1
              ; PROGRAMMSTART BEI $0506

0500      JSR $FE0E   ; TRACK LOESCHEN
0503      JMP $FD9E   ; ZUR JOBSCHLEIFE
0506      LDA #$01    ; TRACKNUMMER
0508      STA $0A     ; IN JOBSPEICHER
050A      LDA #$E0    ; JOBCODE
050C      STA $02     ; UEBERGEBEN
050E WAIT LDA $02     ; RUECKMELDUNG
0510      BMI WAIT    ; ENDE ABWARTEN
0512      RTS         ; PROGRAMMENDE
```

Listing 17. Herstellen eines »21, READ ERROR« auf einer Spur. Startadresse bei $ 0506.

Entspricht der Inhalt des Puffers nicht dem Inhalt auf Diskette, so erhalten wir als Rückmeldung die Nummer 7. Beim LOAD-Befehl entspräche das einem »VERIFY ERROR«.

Übrigens: Es wurde ja schon auf die Notwendigkeit des Initialisierens hingewiesen. Unterbleibt dieser Vorgang, so können Sie anhand der Tabelle 16 schon erkennen, was für eine Meldung Sie bekommen werden. Richtig! Die Nummer 11 wird auf Ihrem Bildschirm erscheinen.

4) Suchen eines Sektors:
Dieses Kommando dient nicht dem Lesen eines Blocks von Diskette. Hier wird lediglich untersucht, ob sich der von Ihnen angegebene Block überhaupt auf Diskette befindet. Ist das nicht der Fall, so erhalten Sie eine »2« als Antwort.

Ihnen ist vielleicht auch schon ein weiterer Vorteil der Job-Schleife aufgefallen: Es erfolgt keine Kontrolle auf »legale« Angaben mehr; das heißt, wenn Sie an den Diskcontroller das Kommando geben, daß er Block 2 auf Track 38 lesen soll, dann tut er dies auch.

Versuchen Sie das einmal mit dem U1-Befehl; hier bekommen Sie als Antwort: »ILLEGAL TRACK OR SEKTOR«, da Track 38 gar nicht existiert.

So groß der Vorteil dieser Nichtkontrolle auch sein mag; sie sollten sich dessen immer bewußt sein, daß der DC auch versuchen würde, auf Track 100 zuzugreifen, wenn dies verlangt werden sollte.

Die Folge wäre hierbei ein Anschlagen des Kopfes an die vordere Laufschienenbegrenzung der Mechanik; eine sicherlich nicht sehr schonende Angelegenheit.

5. Kopf neu positionieren (Bump):
Dieser Befehl hat eine nützliche Funktion, die jedoch auch für eine sicher nicht unerhebliche Menge an verstellten Tonköpfen verantwortlich ist. Kann der DC einen Track nicht identifizieren, so besteht die Möglichkeit, daß der Kopf sich auf einer illegalen Spur befindet. In diesem Fall kann der DC die Position des Kopfes nicht mehr anhand der Blockheader auf jedem Track bestimmen.

Aus diesem Grund passiert folgendes: Der DC fährt den Kopf zurück an den Anschlag, und nach einem »Rattern« erfolgt eine neue Ansteuerung des gewünschten Tracks.

Mit dem Kommando $C0 können Sie ein solches Bump ausführen lassen. Nach dem Bump können Sie den Kopf neu positionieren lassen; der Tonkopf steht ansonsten immer auf Track 1.

6) Maschinenprogramm im Puffer starten:
Mit dem Jobcode $D0 machen Sie intern genau das, was extern mit dem M-E-Befehl funktioniert. Der Unterschied zum M-E-Befehl besteht nur in der Tatsache, daß das Programm, das durch $D0 aufgerufen wird, als Interruptprogramm arbeitet, das heißt es wird in die Job-Schleife mit eingebaut und darf deshalb nicht mit einem RTS enden, da ein JMP zurück in die Jobschleife erfolgen muß.

Wie Sie aus einem solchen Programm zurückspringen, wird später noch erläutert.

7) Programm im Puffer starten, nachdem das Laufwerk hochgefahren ist:
Den letzten Befehl werden wir kaum benutzen, da ihm eine Eigenschaft fehlt, die wir dringend benötigen. Wollen wir nämlich ein Programm in der Job-Schleife starten, so werden wir meistens Schreib- oder Lesezugriffe auf die Diskette ausführen. Dies ist jedoch mit $D0 nicht möglich, da das Laufwerk stillsteht.

Der Befehl $E0 hat nun folgende Auswirkungen: Erkennt der DC den Jobcode, so wird das Laufwerk angefahren und die Hardware auf Diskettenzugriff vorbereitet. Mit Hilfe dieses Befehls ist es also möglich, direkt auf die Diskette zuzugreifen, was in einem eigenen Maschinenprogramm erfolgt.

Auch hier muß das Programm mit einem JMP-Befehl beendet werden, da ein Rücksprung in die Job-Schleife erfolgen soll.

Wichtig ist noch, daß das Programm, das mit $D0 oder $E0 gestartet werden soll, immer am Anfang des entsprechenden Puffers stehen muß. Sollen also Programmteile aufgerufen werden, die an höheren Adressen, als $xx00 (xx=03 bis 07) stehen, so müssen diese über Sprungbefehle aufgerufen werden.

Wie schreibt das DOS auf Diskette?

Mit $E0 werden wir uns in unserem Kurs noch öfters beschäftigen, da er die Grundlage der Diskettenzugriffe darstellt (er wird auch vom DOS für das Formatieren angewendet).

Eine Sache dürfen Sie aber auch beim Jobcode $E0 nicht vergessen, nämlich Track- und Sektornummer anzugeben. Es wird, wie schon erwähnt, das Laufwerk betriebsbereit gemacht. Dazu gehört aber auch das Positionieren des Tonkopfes auf die richtige Spur.

Wir haben jetzt die Möglichkeit, ein Maschinenprogramm im Pufferspeicher der Floppy abzulegen und dort zu starten. Unsere Jobcodes erlauben es uns außerdem, direkt in den Ablauf der Job-Schleife einzugreifen und die Diskette sozusagen »von Hand« zu manipulieren.

Als letztes fehlen uns jetzt nur noch die Kenntnisse über den direkten Zugriff auf den Schreib-/Lesekopf der Floppy, so daß wir einzelne Bits ohne Umwege und ohne irgendeine Einschränkung durch die Blockstruktur der Diskette direkt auf die Magnetschicht schreiben können. Mit diesem Problem, das eigentlich gar keines ist, wollen wir uns jetzt beschäftigen. Dazu ein paar Bemerkungen zur Organisation der Schreib-/Leseelektronik der Floppy 1541.

Die Bytes werden zwar auf Diskette in serieller Bitfolge abgelegt; dieses Problem braucht uns jedoch gar nicht weiter zu beschäftigen. Der VIA 6522, der für uns die Elektronik steuert, kann nämlich von uns wie eine Speicherstelle behandelt werden. Senden wir also ein Byte an den VIA 6522, so geschieht das Schreiben auf Diskette vollautomatisch, so daß uns diese Sache nicht weiter beschäftigen soll.

Das einzige Problem, das sich bei der ganzen Angelegenheit stellt, ist die Frage des Timing. Immerhin benötigt

der Schreib- oder Lesevorgang eine gewisse Zeit, das heißt wenn wir beispielsweise Daten vom Tonkopf lesen wollen, muß uns der DC erst mitteilen, wann das nächste Byte fertig eingelesen ist und zur Ausgabe bereitsteht.

Zur Steuerung dieses Timings wird in der Floppy 1541 das V-(Overflow-)Flag des Prozessorstatusregisters benutzt. Der Mikroprozessor 6502 hat nämlich den Vorteil, daß dieses Flag extern beeinflußt werden kann.

Die Regel sieht also folgendermaßen aus: Hat die Lese-Elektronik ein Byte vollständig eingelesen, so wird das V-Flag auf »1« gesetzt. Genauso verhält es sich mit dem Schreiben: Wurde das gegebene Byte komplett auf Diskette geschrieben, so erfolgt ebenfalls ein Setzen des V-Flags.

Das einzige, das der Programmierer nie vergessen darf, ist, daß das V-Flag nach dem Erkennen »von Hand« wieder auf »0« gesetzt werden muß, damit keine Fehlinformation erfolgen kann.

Die Speicherstelle, die für Schreib- und Lesebetrieb zuständig ist, ist Port A des DC mit der Adresse $1C01.

Endlich kommt die Praxis

So, nachdem wir nun so ziemlich alle Voraussetzungen zum Programmieren haben, soll es jetzt endlich mit der praktischen Anwendung unseres Wissens losgehen. Das Werkzeug, das wir jetzt benötigen, besteht aus einem komfortablen Monitor mit »Miniassembler«. Da die Floppy-Programme, die zum Beispiel Fehler auf Diskette bringen, relativ kurz sind, ist es am besten, wenn wir einen Monitor in den Bereich ab $C000 laden (zum Beispiel den SMON) und uns anschließend den Bereich ab $8000 für unsere Anwendungen sichern:

POKE 56,127: POKE 52,127: NEW (oder CLR)

Wir legen also unsere kleinen Maschinenprogramme ab $8000 ab und senden diese jeweils mit einem Basic-Programm zur 1541, wo wir sie dann ausführen.

Achtung: Bei einem Reset wird der Speicher der Floppystation gelöscht. Es ist also empfehlenswert, die Programme vor jedem Neustart wieder in den Pufferspeicher des 1541-Laufwerks zu schreiben.

Error Nummer 21 auf Diskette

Ein früher beliebter Programmschutz war das Aufbringen von Errors auf Diskette. Diese konnten von den »alten« Kopierprogrammen nicht übernommen werden. Das geschützte Programm brauchte also nur einen definierten

```
  AUF SCHREIBEN UMSCHALTEN

LDA $1COC    ; PCR
AND #$1F     ; AUF SCHREIBMODUS
ORA #$DO     ;
STA $1COO    ; UMSCHALTEN
LDA #$FF     ;
STA $1CO3    ; PORT A AUF AUSGANG

  AUF LESEN UMSCHALTEN (AUCH JSR $FEOO)

LDA $1COC    ; PCR
ORA #$EO     ; AUF LESEMODUS
STA $1COC    ; UMSCHALTEN
LDA #$OO     ;
STA $1CO3    ; PORT A AUF EINGANG
```

Listing 18a. Ein READ ERROR 22 wird erzeugt (in einem beliebigen Sektor)

```
5 REM PROGRAMM ZUM ERZEUGEN EINES               <209>
6 REM     22, READ ERROR                        <197>
7 :                                             <239>
10 POKE 56,31:POKE 52,31:CLR:OPEN 1,8,15,"
   I"                                           <192>
20 FOR X=0 TO 80:READ A:POKE 32768+X,A:NEX
   T                                            <105>
30 INPUT"{CLR,DOWN,SPACE}TRACK FUER ERROR
   22";T                                        <222>
40 INPUT"{DOWN}SEKTOR FUER ERROR 22";S          <006>
50 POKE 32777,T:POKE 32834,T:POKE 32781,S       <202>
60 RESTORE                                      <110>
70 FOR X=0 TO 80:PRINT#1,"M-W"CHR$(X)CHR$(
   5)CHR$(1)CHR$(PEEK(X+32768)):NEXT            <032>
80 PRINT:PRINT:PRINT"PROGRAMM STARTET"          <052>
90 PRINT#1,"M-E"CHR$(64)CHR$(5):CLOSE 1:EN
   D                                            <056>
100 DATA 165,18,133,22,165,19,133,23,169,3
    5,133,24,169,1,133,25,32,39                 <211>
110 DATA 245,32,86,245,173,12,28,41,31,9,1
    92,141,12,28,169,255,141,3,28               <219>
120 DATA 169,85,141,1,28,80,254,184,80          <108>
130 DATA 254,184,80,254,184,32,0,254,76         <156>
140 DATA 158,253,234,234,234,234,234,234        <103>
150 DATA 234,234,169,35,133,10,169,224,133
    ,2,165,2,48,252,96,0,0,0                    <224>
```

Listing 18b. Herstellen eines »22, READ ERROR« (Assemblerprogramm)

Fehler auf Diskette abzufragen und bei Nichtvorhandensein »auszusteigen«. Wenn Sie sich die Tabelle der Fehlermeldungen im Commodore-Handbuch zur Floppy 1541 ansehen, werden Sie sehr schnell erkennen, daß es für jeden kleinen Defekt eine eigene Fehlernummer gibt. Betrachten Sie jetzt Tabelle 16 dieser Folge, so können Sie dort ablesen, welche Rückmeldung des DC welche Fehlermeldung an den Computer zur Folge hat.

Wir wollen uns einmal den Fehler mit der Nummer 21 ansehen. Er tritt dann auf, wenn die Floppystation versucht, einen Track zu lesen, auf diesem jedoch keine SYNC-Markierungen findet. Das ist zum Beispiel bei einer unformatierten oder beschädigten Diskette der Fall.

Unser kleines Programm in Listing 17 werden Sie vom Prinzip sehr schnell durchschauen. Es macht nichts weiter, als einen bestimmten Track auf Diskette mit lauter $55 (binär: 01010101) zu überschreiben. Das hat zur Folge, daß alle SYNC-Markierungen gelöscht werden und ein Fehler »21« ist die Folge, wenn ein Zugriff stattfinden soll.

Für unsere Versuche sollten Sie eine leere, neuformatierte Diskette verwenden, die Sie sich speziell für unsere Experimente aufheben. Geben Sie also einmal das Programm in Listing 17 ein und starten Sie es anschließend (leere Diskette einlegen!).

Versuchen Sie nun, den Track 1 Ihrer Diskette später einmal zu lesen, so wird sich die Floppy mit einem »21, READ ERROR« dafür bedanken.

Wie Sie sehen, ist ein Fehler 21 recht einfach zu erzeugen, da sich dieser über einen gesamten Track erstreckt (alle Informationen werden gelöscht).

Schwieriger wird es bei anderen Fehlern, die beispielsweise nur in einzelnen Blöcken vorkommen, wobei einige davon (20, 22) auch auf einen gesamten Track geschrieben werden können. Es sind dies die Fehler mit den Nummern 23, 24, 27, 28 und 29.

Um solche Fehler zu erzeugen, muß jeweils der zu zerstörende Sektor abgetastet werden, bis die richtige Stelle für den Eingriff gefunden wird. Damit Sie die wichtigen Routinen zur Arbeit innerhalb der Job-Schleife ebenfalls aufrufen können, sind in Tabelle 14 ein paar wichtige Unterprogramme des DOS mit den geforderten Parametern aufgeführt.

Einen Error 22 beispielsweise würden Sie dadurch herstellen, daß Sie die Routine zum Finden des Datenblocks

```
0500         LDA $12     ; ID 1 HOLEN
0502         STA $16     ; UND UEBERNEHMEN
0504         LDA $13     ; ID 2 HOLEN
0506         STA $17     ; UND UEBERNEHMEN
0508         LDA #$23    ; TRACKNUMMER
050A         STA $18     ; UEBERNEHMEN
050C         LDA 01      ; SEKTORNUMMER
050E         STA $19     ; UEBERNEHMEN
0510         JSR $F527   ; BLOCKHEADER HOLE
                           N
0513         JSR $F556   ; AUF 'SYNC' WARTE
                           N
0516         LDA $1C0C   ; PCR
0519         AND #$1F    ; AUF SCHREIBEN
051B         ORA #$C0    ; UMSCHALTEN
051D         STA $1C0C   ;
0520.        LDA #$FF    ; PORT A AUF AUSGA
                           NG
0522         STA $1C03   ; UMSCHALTEN
0525         LDA #$55    ; FALSCHWERT
0527         STA $1C01   ;
052A W1      BVC W1      ; SCHREIBEN
052C         CLV         ;
052D W2      BVC W2      ; SCHREIBEN
052F         CLV         ;
0530 W3      BVC W3      ; SCHREIBEN
0532         CLV         ;
0533         JSR $FE00   ; PCR AUF LESEN
0536         JMP $FD9E   ; ZUR JOBSCHLEIFE
0539         NOP         ;
053A         NOP         ;
053B         NOP         ;
053C         NOP         ;
053D         NOP         ;
053E         NOP         ;
053F         NOP         ;
0540         NOP         ;
0541         LDA #$23    ; TRACKNUMMER
0543         STA $0A     ; IN JOBSPEICHER
0545         LDA #$E0    ; JOBCODE
0547         STA $02     ; UEBERGEBEN
0549 WAIT    LDA $02     ; RUECKMELDUNG
054B         BMI WAIT    ; WARTEN AUF ENDE
054D         RTS         ; PROGRAMMENDE
```

Listing 18c. Herstellen eines »22, READ ERROR« (Assemblerprogramm)

```
5 REM PROGRAMM ZUM ERZEUGEN EINES            <209>
6 REM     23, READ ERROR                     <213>
7 :                                          <239>
10 POKE 56,31:POKE 52,31:CLR:OPEN 1,8,15,"
   I"                                        <192>
20 FOR X=0 TO 80:READ A:POKE 32768+X,A:NEX
   T                                         <105>
30 INPUT"{CLR,DOWN,SPACE}TRACK FUER ERROR
   23";T                                     <254>
40 INPUT"{DOWN}SEKTOR FUER ERROR 23";S       <038>
50 POKE 32777,T:POKE 32834,T:POKE 32781,S    <202>
60 RESTORE                                   <110>
70 FOR X=0 TO 80:PRINT#1,"M-W"CHR$(X)CHR$(
   5)CHR$(1)CHR$(PEEK(X+32768)):NEXT         <032>
80 PRINT:PRINT:PRINT"PROGRAMM STARTET"       <052>
90 PRINT#1,"M-E"CHR$(64)CHR$(5):CLOSE 1:EN
   D                                         <056>
100 DATA 165,18,133,22,165,19,133,23,169,3
    5,133,24,169,0,133,25,32,39              <209>
110 DATA 245,32,86,245,162,0,202,208,253     <229>
120 DATA 173,12,28,41,31,9,192,141,12,28,1
    69,255,141,3,28,169,85,141,1             <121>
130 DATA 28,80,254,184,80,254,184,80,254,1
    84,32,0,254,76,158,253,234,234           <084>
140 DATA 234,169,35,133,10,169,224,133,2,1
    65,2,48,252,96,0,0,0                     <083>
```

Listing 19a. Ein READ ERROR 23 wird erzeugt

aufrufen. Diese kehrt bei gefundenem Datenblock mit RTS zurück. Jetzt schalten Sie auf Schreiben um (in Bild 6 dargestellt) und bringen ein paar Byte ohne Konzept auf die Diskette. Versucht der DC, diesen Datenblock später einmal zu lesen, so erfolgt ein Fehler 22, da Sie die Datenblockkennung, die direkt hinter der SYNC-Markierung steht, zerstört haben.

Wollen Sie einen Fehler mit der Nummer 23, dann ist es erforderlich, daß Sie den Vorspann des Datenblocks überspringen und erst inmitten der gespeicherten Daten einen Schreibzugriff durchführen. Durch diesen Zugriff, der in der Prüfsumme am Blockende natürlich nicht verzeichnet wird, folgt die Meldung »23, READ ERROR«, als Zeichen eines Prüfsummenfehlers.

Listing 18a und 19a zeigen Ihnen Programme, die einen Error 22 und einen Error 23 erzeugen (Listing 18b und 19b sind die zugehörigen Quellprogramme).

Der Vorteil eines Fehlers mit der Nummer 23 ist, daß die Daten in der Regel schon im Puffer stehen, bevor der Fehler erkannt wird, das heißt Sie können einen Datenblock auf Diskette gezielt mit einem Fehler versehen, obwohl dieser noch lesbare Inhalte enthält.

Fehler als Kopierschutz

Die eben besprochenen Fehler auf Diskette eignen sich hervorragend für einen Kopierschutz. Am wirkungsvollsten sind dabei mit Sicherheit solche Fehler, die zusätzlich noch Daten enthalten. Es gibt nämlich schon eine ganze Menge von Programmen, die Fehler übernehmen und auf der Kopie wieder simulieren.

Soweit zu Fehlern. Haben Sie schon einmal etwas von »Killertracks« gehört? Dieses anschauliche Wort steht für die Manipulation eines Tracks, der sämtliche Sicherheitseinrichtungen des DOS durcheinanderbringt.

Vielleicht hatten Sie schon einmal eine Diskette in Ihren Händen, die folgendes »Phänomen« aufzeigte: Wenn Sie versuchten, einen Block auf einer bestimmten Spur zu lesen, ist der Schreib-/Lesekopf der Floppystation ordnungsgemäß auf den Track positioniert worden. Danach hat der DC mit dem Lesen des Blocks angefangen und – nicht mehr aufgehört. Anders ausgedrückt: Die Floppy 1541 las und las ...

Die Spur, die Sie da versucht haben zu lesen, hat grundsätzlich dafür gesorgt, daß sich die Diskettenstation »aufhängte«. Daß es sich hier um den schon angesprochenen »Killertrack« handelte, brauche ich kaum noch zu erwähnen. Aber, wie stellt man eine solche »Falle« her? Was ist mit dem Track passiert, daß der DC völlig »aus dem Häuschen« gerät? Die Antwort sehen Sie in Listing 20. Dieses kleine Programm stellt einen solchen »Killertrack« her. Des Rätsels Lösung ist eigentlich ganz einfach: Die gesamte Spur besteht aus einer einzigen SYNC-Markierung. Da die SYNC-Markierung von der Lese-/Schreibelektronik speziell verarbeitet wird, verzögert sich die Arbeit des DC gewaltig, wenn eine solche »Dauer-SYNC-Markierung« auftritt.

Da die Floppystation bei Fehlern bis zu über 200mal versucht, einen Block zu lesen, dehnt sich der Zeitraum, den sie bei Verzögerungen benötigt, stark aus. Bei Killertracks braucht die Diskettenstation pro Leseversuch eine Unmenge an Zeit, was sich auch im langsamen Blinkrhythmus der LED am Laufwerk zeigt.

Allein schon an den kleinen Anwendungen können Sie erkennen, wie vielseitig und vielfältig die Möglichkeiten sind, die einem in der Programmierung offenstehen. Wenn Sie intensiv mit der Floppystation arbeiten, werden Sie bald schon neue Anwendungsmöglichkeiten kennenlernen.

Aus der Floppy 1541 läßt sich noch eine Menge herausholen, wie wir noch feststellen werden, wobei der Kopierschutz von Disketten sicher nur einen kleinen Teil der vielfältigen Möglichkeiten darstellt.

Wie jedem Floppy-Besitzer bekannt ist, muß eine Diskette vor dem ersten Speichern von Daten formatiert werden. Wie eine Diskette nach einem solchen Formatiervorgang aussieht, wurde schon besprochen.

Uns soll nun interessieren, was während des Formatierens so alles in der Floppystation passiert, und warum die 1541 so lange für einen eigentlich sehr einfachen Vorgang benötigt.

Zur Wiederholung: Beim Formatieren werden vom DOS alle wichtigen Markierungen auf die Diskette gebracht und außerdem sämtliche Sektoren in ihrer späteren Form angelegt.

Der Vorgang des Formatierens verwendet zu seiner Ausführung einen uns schon bekannten Jobcode, nämlich $E0.

Bevor das DOS den eigentlichen Formatiervorgang startet, wird ab $0600 (also im Puffer 3) ein Sprungbefehl abgelegt: JMP $FAC7.

Dieser Sprungbefehl ist eine Art Vektor, der im RAM liegt und somit verändert werden kann. Er bietet dem Benutzer die Möglichkeit, eine eigene Routine einzubauen, die dann bei jedem Track-Wechsel angesprungen wird, um so einige wirksame Manipulationen an der Formatierung vorzunehmen, indem zum Beispiel Werte in der Zero-Page verändert werden, doch dazu später. Üblicherweise zeigt dieser Vektor direkt auf eine Jobroutine, die für das Formatieren zuständig ist. Diese Routine wird nun vom Hauptprogramm mit dem Jobcode $E0, der in Speicherstelle $03 geschrieben wird, aufgerufen.

Formatieren in der Job-Schleife

Am Anfang der Jobroutine steht nun die Abfrage, ob schon mindestens ein Track formatiert wurde oder ob dieser Einsprung der allererste ist. Ist dieser Einsprung der erste, so werden alle Parameter für den Steppermotor gesetzt; danach erfolgt ein Rücksprung in die Job-Schleife. Hier wird der Schreib-/Lesekopf nun 45 (!) Tracks zurückgefahren, was sich in jenem charakteristischen Rattern der Floppystation äußert.

Nun, können Sie sagen, es würde auch reichen, wenn der Kopf nur 35 oder 40 Spuren zurücktransportiert würde. In der Tat ist der Wert 45 sehr hoch. Man muß aber bedenken, daß es passieren kann, daß der Schreib-/Lesekopf der Floppystation durch eine defekte Diskette oder einen Pro-

```
0500      LDA $12     ; ID 1 HOLEN
0502      STA $16     ; UND UEBERNEHMEN
0504      LDA $13     ; ID 2 HOLEN
0506      STA $17     ; UND UEBERNEHMEN
0508      LDA #$23    ; TRACKNUMMER
050A      STA $18     ; UEBERNEHMEN
050C      LDA 00      ; SEKTORNUMMER
050E      STA $19     ; UEBERNEHMEN
0510      JSR $F527   ; BLOCKHEADER HOLE
                        N
0513      JSR $F556   ; AUF 'SYNC' WARTE
                        N
0516      LDX #$00    ; WARTEN, UM IN
0518 LOOP DEX         ; DEN DATENBLOCK
0519      BNE LOOP    ; ZU KOMMEN
051B      LDA $1C0C   ;
051E      AND #$1F    ; PCR AUF SCHREIBE
                        N
0520      ORA #$C0    ; UMSCHALTEN
0522      STA $1C0C   ;
0525      LDA #$FF    ; PORT A AUF
0527      STA $1C03   ; AUSGANG STELLEN
052A      LDA #$55    ; FALSCHWERT
052C      STA $1C01   ; IN PUFFER SCHREI
                        BEN
052F W1   BVC W1      ; WARTEN AUF READY
0531      CLV         ; FLAG LOESCHEN
0532 W2   BVC W2      ; WARTEN AUF READY
0534      CLV         ; FLAG LOESCHEN
0535 W3   BVC W3      ; WARTEN AUF READY
0537      CLV         ; FLAG LOESCHEN
0538      JSR $FE00   ; AUF LESEN SCHALT
                        EN
053B      JMP $FD9E   ; ZUR JOBSCHLEIFE
053E      NOP         ;
053F      NOP         ;
0540      NOP         ;
0541      LDA #$23    ; TRACKNUMMER
0543      STA $0A     ; IN JOBSPEICHER
0545      LDA #$E0    ; JOBCODE
0547      STA $02     ; UEBERGEBEN
0549 WAIT LDA $02     ; RUECKMELDUNG
054B      BMI WAIT    ; WARTEN AUF ENDE
054D      RTS         ; PROGRAMMENDE
```

Listing 19b. Herstellen eines »23, READ ERROR« (Assemblerprogramm)

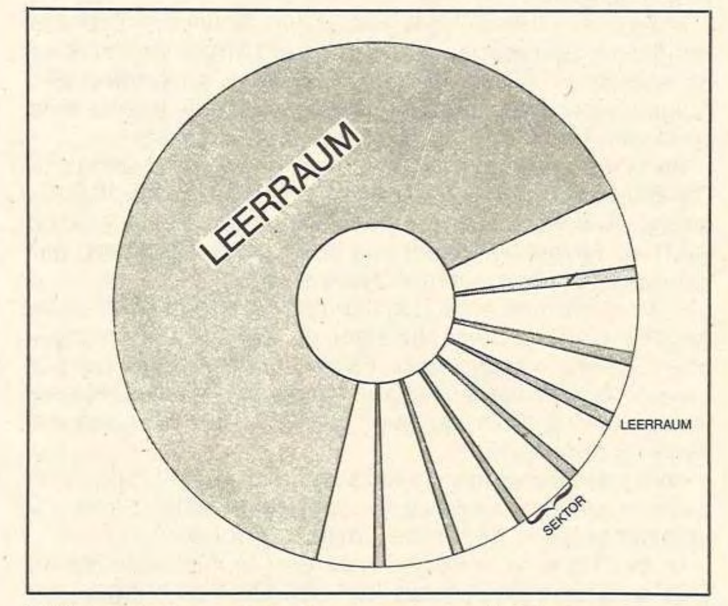


Bild 6. Verteilung ohne Abmessung der Abstände

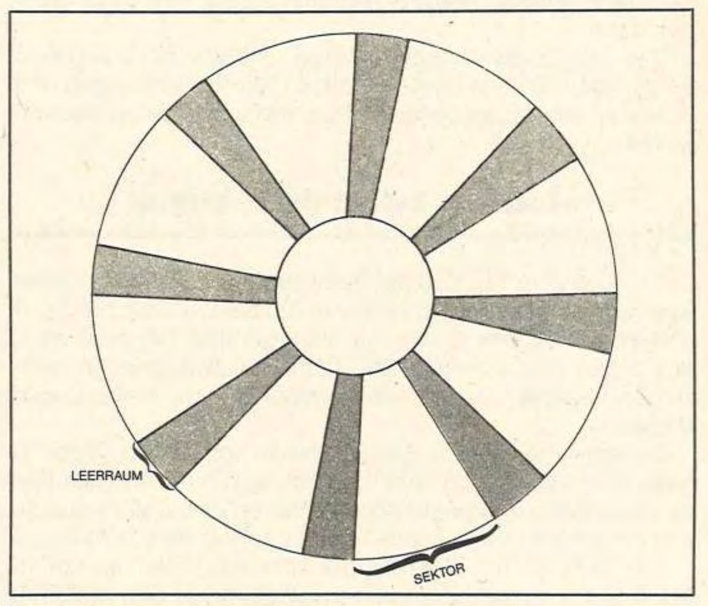


Bild 7. Symmetrische Verteilung der Sektoren

grammierfehler zu weit nach innen gefahren und beispielsweise auf Track 42 am Anschlag gelandet ist und daß ein Zurückfahren um 40 Tracks einfach nicht ausreicht, um den Kopf richtig zu positionieren.

Der Wert von 45 Tracks enthält also eine Sicherheitsreserve, die ein Positionieren auf Spur 1 mit Sicherheit ermöglicht.

Wurde der Kopf also auf Track 1 positioniert, so erfolgt erneut ein Einsprung in die Formatierungsroutine; eine Speicherstelle zeigt jetzt an, daß der Kopf auf Track 1 positioniert wurde und das Formatieren starten kann.

Jetzt wird noch geprüft, ob auf die nächste Spur umgeschaltet werden soll, da die aktuelle bereits formatiert wurde (wenn ja, erfolgt wieder ein Einsprung in die Job-Schleife, um das Nötige zu tun).

Diese Abfragen am Anfang der Formatierungsroutine scheinen umständlich und überflüssig zu sein; das Gefühl täuscht jedoch. Wir dürfen ja nicht vergessen, daß die Routine immer nur jeweils einen Track formatiert und danach zur Job-Schleife zurückkehrt, damit der Tonkopf weitergeführt werden kann. Wir haben also gewissermaßen eine Endlosschleife, die nur durch die Feststellung, daß Spur 35 fertig formatiert wurde, beendet wird.

Ausmessen einer Spur

Jetzt haben wir aber endlich alle Voraussetzungen zum Formatieren eines Tracks erfüllt und wollen an die Arbeit gehen. Der Abschnitt, der jetzt besprochen wird, ist übrigens

```
0500      JSR $FDA3 ; TRACK LOESCHEN
0503      JMP $FD9E ; ZUR JOBSCHLEIFE
0506      LDA #$01  ; TRACKNUMMER
0508      STA $0A   ; IN JOBSPEICHER
050A      LDA #$E0  ; JOBCODE
050C WAIT STA $02   ; UEBERGEBEN
050E      LDA $02   ; RUECKMELDUNG
0510      BMI WAIT  ; ENDE ABWARTEN
0512      RTS       ; PROGRAMMENDE
```

Listing 20. Ein »Killertrack« wird erzeugt. Startadresse bei $0506.

für die langwierige Formatierung verantwortlich und sorgt für die ausgedehnten Wartezeiten.

Bevor die SYNC-Markierungen und Sektoren auf eine Spur geschrieben werden, wird diese Spur vom DOS »ausgemessen«.

Das Betriebssystem der Floppy 1541 »weiß« im Normalfall genau, wie viele Bytes für die SYNC-Markierungen und Sektoren einer Spur benötigt, beziehungsweise verbraucht werden.

»Löcher« zwischen Sektoren

Jetzt ist es aber so, daß die Sektoren nicht genau auf jede Spur abgemessen sind; vielmehr hat die Diskette pro Spur eine etwas höhere Kapazität, als eigentlich benötigt wird. Aus dieser Tatsache folgt natürlich, daß zwischen den einzelnen Sektoren »Leerstellen« entstehen, die keine Daten enthalten.

Da jetzt aber die Länge der Tracks von außen (Track 1) nach innen (Track 35) kontinuierlich abnimmt, werden diese Leerstellen immer kleiner; wir haben also unterschiedliche Anzahlen von »Leer-Bytes« zwischen den Sektoren.

Das DOS ist nun bestrebt, die Sektoren jeder Spur möglichst symmetrisch anzuordnen, also immer den gleichen Abstand zwischen zwei Sektoren eines Tracks zu haben. Bild 6 zeigt, was passiert, wenn keine vorherige Ausmessung stattfindet.

Um das Ziel einer »symmetrisch« formatierten Diskette (Bild 7) zu erreichen, stellt das DOS durch einige komplizierte Schreib- und Lesevorgänge das Verhältnis zwischen benötigtem und vorhandenem Platz einer Spur fest. Aus diesem Verhältnis kann nun anhand einer einfachen Rechnung festgestellt werden, wieviel Platz zwischen den einzelnen Sektoren freigelassen werden muß.

Formatieren mit Variationen

Nachdem diese komplizierte Vermessung stattgefunden hat, die mehrere Diskettenumdrehungen und damit Zeit erfordert, beginnt nun das eigentliche Formatieren der Diskette, das mit allem Drum und Dran normalerweise nicht mehr als eine ⅕ Sekunde für einen Track benötigt.

Das Anlegen der Sektoren im Puffer

Bevor geschrieben werden kann, müssen die Sektoren erst einmal im Pufferspeicher der Floppy 1541 hergestellt werden. Da sich die einzelnen Sektoren nur durch deren Header unterscheiden, reicht das Anlegen der Blockheader; die Inhalte der Datenblöcke sind bei jedem Sektor gleich und bestehen aus dem schon bekannten Muster $4B gefolgt von 255 $01-Byte.

Die Blockheader werden alle in einem Pufferspeicher ($0300-$03FF) abgelegt; der Inhalt der Datenblöcke steht ab $0500 bis $05FF.

Schreiben eines Tracks auf Diskette

So, alle Vorarbeiten wären jetzt abgeschlossen. Wir können mit dem Schreiben auf Diskette beginnen. Zuerst wird der Diskcontroller auf Schreibmodus gestellt und die Spur der Diskette gelöscht.

Der gesamte Spurinhalt wird nun während einer einzigen Diskettenumdrehung (⅕ Sekunde) auf die Diskette gebracht, wobei zuerst die SYNC-Markierung für den Blockheader, danach der Blockheader selbst, geschrieben werden. Nach einer Lücke von 9 Byte folgt die SYNC-Markierung des Datenblocks mit dem zugehörigen Daten-Byte. Den Abschluß eines Sektors bildet der schon erwähnte »Leerraum«, der aus der vorher errechneten Anzahl von Bytes besteht.

Zur Sicherheit erfolgt nach dem Schreiben eine Verify-Routine, die auf eventuelle Disketten- oder Schreibfehler kontrolliert und bei deren Auftreten einen »24, READ ERROR« ausgibt.

Mit dieser letzten Maßnahme ist eine Spur einer Diskette fertig formatiert worden, und es wird auf Erreichen der Spur 35 abgefragt. Wurde Spur 35 formatiert, so werden alle Flags für das Formatieren zurückgesetzt, die Jobschleife verlassen und ins Hauptprogramm zurückgekehrt.

Im Hauptprogramm wird nun auf Track 18 positioniert. Die BAM der Diskette wird hergestellt und in Block 18,0 abgelegt. Anschließend wird noch der erste Directory-Block (18,1) mit Nullen vollgefüllt und ebenfalls gespeichert, womit das Formatieren abgeschlossen wäre.

Formatiert man eine Diskette nur kurz, das heißt ohne Angabe einer ID beim N-Befehl, so werden alle anfänglichen Schritte weggelassen. Es wird in diesem Fall nur auf das richtige Formatkennzeichen in der BAM ($41/65/A) kontrolliert und danach der eben beschriebene Vorgang auf Track 18 durchgeführt.

Nun wäre unser Floppy-Kurs natürlich kein Floppy-Kurs, wenn wir unsere neu gewonnenen theoretischen Kenntnisse nicht sofort in die Praxis umsetzen wollten.

In der Tat kann man mit Hilfe der Formatierroutine im DOS einige nette »Scherze« auf eine Diskette bringen, die entweder dem Spieltrieb oder dem Softwareschutz dienen.

```
Floppyprogramm zum Disk-Format-System
    1985 by KOSS
0500 ea       nop
0501 a5 0a    lda $0a      Tracknummer aus Jobspeicher
0503 c9 24    cmp #$24     größer als 35?
0505 90 07    bcc $050e    nein
0507 a9 12    lda #$12     ja: 18 als Anzahl der Sektoren
0509 85 43    sta $43      festlegen
050b 4c 13 05 jmp $0513
050e 20 4b f2 jsr $f24b    Anzahl der Sektoren holen
0511 85 43    sta $43      und merken
0513 a9 00    lda #$00
0515 85 1b    sta $1b      Sektorzähler setzen
0517 a0 00    ldy #$00
0519 a2 00    ldx #$00
051b a5 39    lda $39      Kennzeichen $08 für Blockheader
051d 99 00 03 sta $0300,y
0520 c8       iny
0521 c8       iny
0522 a5 1b    lda $1b      Sektornummer
0524 99 00 03 sta $0300,y
0527 c8       iny
0528 a5 0a    lda $0a      Tracknummer
052a 99 00 03 sta $0300,y
052d c8       iny
052e a5 13    lda $13      ID 2
0530 99 00 03 sta $0300,y
0533 c8       iny
0534 a5 12    lda $12      ID 1
0536 99 00 03 sta $0300,y
0539 c8       iny
053a a9 0f    lda #$0f
053c 99 00 03 sta $0300,y  Lücke lassen
053f c8       iny
0540 99 00 03 sta $0300,y
0543 c8       iny
0544 a9 00    lda #$00
0546 59 fa 02 eor $02fa,y  Prüfsumme bilden
0549 59 fb 02 eor $02fb,y
054c 59 fc 02 eor $02fc,y
054f 59 fd 02 eor $02fd,y
0552 99 f9 02 sta $02f9,y  und abspeichern
0555 e6 1b    inc $1b
0557 a5 1b    lda $1b      Sektorzähler erhöhen
0559 c5 43    cmp $43      schon Maximalzahl?
055b 90 be    bcc $051b    nein, weitermachen
055d a9 03    lda #$03
055f 85 31    sta $31
0561 98       tya
0562 48       pha
0563 8a       txa
0564 9d 00 07 sta $0700,x  Datenblock mit $00 füllen
0567 e8       inx
0568 d0 fa    bne $0564
056a 20 30 fe jsr $fe30
056d 68       pla
056e a8       tay
056f 88       dey
0570 20 e5 fd jsr $fde5
0573 20 f5 fd jsr $fdf5
0576 a9 07    lda #$07
0578 85 31    sta $31
057a 20 e9 f5 jsr $f5e9    Prüfsumme für Datenblock
057d 85 3a    sta $3a      abspeichern
057f 20 8f f7 jsr $f78f
0582 a9 00    lda #$00     Sektorzähler setzen
0584 85 32    sta $32
0586 20 0e fe jsr $fe0e    Track löschen
0589 a9 ff    lda #$ff
058b 8d 01 1c sta $1c01    SYNC schreiben
058e a2 05    ldx #$05
0590 50 fe    bvc $0590
0592 b8       clv
0593 ca       dex
0594 d0 fa    bne $0590
0596 a2 0a    ldx #$0a
0598 a4 32    ldy $32
059a 50 fe    bvc $059a    Blockheader schreiben
059c b8       clv
059d b9 00 03 lda $0300,y
05a0 8d 01 1c sta $1c01
05a3 c8       iny
05a4 ca       dex
05a5 d0 f3    bne $059a
05a7 a2 09    ldx #$09
05a9 50 fe    bvc $05a9    Lücke von 9 Bytes lassen
05ab b8       clv
05ac a9 55    lda #$55
05ae 8d 01 1c sta $1c01
05b1 ca       dex
05b2 d0 f5    bne $05a9
05b4 a9 ff    lda #$ff
05b6 a2 05    ldx #$05
05b8 50 fe    bvc $05b8    SYNC-Markierung für Datenblock
05ba b8       clv
05bb 8d 01 1c sta $1c01
05be ca       dex
05bf d0 f7    bne $05b8
05c1 a2 bb    ldx #$bb
05c3 50 fe    bvc $05c3
05c5 b8       clv
05c6 bd 00 01 lda $0100,x
05c9 8d 01 1c sta $1c01
05cc e8       inx
05cd d0 f4    bne $05c3
05cf a0 00    ldy #$00
05d1 50 fe    bvc $05d1    Datenblock schreiben
05d3 b8       clv
05d4 b1 30    lda ($30),y
```

```
05d6 8d 01 1c sta $1c01
05d9 c8       iny
05da d0 f5    bne $05d1
05dc a9 55    lda #$55
05de a2 08    ldx #$08     Lücke nach Sektor mit fester
05e0 50 fe    bvc $05e0    Länge von 8 Bytes schreiben
05e2 b8       clv
05e3 8d 01 1c sta $1c01
05e6 ca       dex
05e7 d0 f7    bne $05e0
05e9 a5 32    lda $32
05eb 18       clc
05ec 69 0a    adc #$0a
05ee 85 32    sta $32
05f0 c6 1b    dec $1b      schon alle Sektoren?
05f2 d0 95    bne $0589    nein, weitermachen
05f4 50 fe    bvc $05f4
05f6 b8       clv
05f7 50 fe    bvc $05f7
05f9 b8       clv
05fa 20 00 fe jsr $fe00    auf Lesen umschalten
05fd a9 c8    lda #$c8     200 Leseversuche
05ff 85 1f    sta $1f
0601 a9 00    lda #$00
0603 85 30    sta $30
0605 a9 03    lda #$03
0607 85 31    sta $31
0609 a5 43    lda $43
060b 85 1b    sta $1b      Sektorzähler
060d 20 56 f5 jsr $f556    auf SYNC-Signal warten
0610 a2 0a    ldx #$0a
0612 a0 00    ldy #$00
0614 50 fe    bvc $0614
0616 b8       clv
0617 ad 01 1c lda $1c01
061a d1 30    cmp ($30),y  Daten vergleichen
061c d0 0e    bne $062c
061e c8       iny
061f ca       dex
0620 d0 f2    bne $0614
0622 18       clc
0623 a5 30    lda $30
0625 69 0a    adc #$0a
0627 85 30    sta $30
0629 4c 35 06 jmp $0635
062c c6 1f    dec $1f      Zähler vermindern
062e d0 d1    bne $0601
0630 a9 06    lda #$06
0632 4c d3 fd jmp $fdd3    24, READ ERROR
0635 20 56 f5 jsr $f556    SYNC-Signal abwarten
0638 a0 bb    ldy #$bb
063a 50 fe    bvc $063a
063c b8       clv
063e ad 01 1c lda $1c01
0640 d9 00 01 cmp $0100,y
0643 d0 e7    bne $062c
0645 c8       iny
0646 d0 f2    bne $063a
0648 a2 fc    ldx #$fc
064a 50 fe    bvc $064a
064c b8       clv
064d ad 01 1c lda $1c01
0650 d9 00 07 cmp $0700,y  Datenblock testen
0653 d0 d7    bne $062c
0655 c8       iny
0656 ca       dex
0657 d0 f1    bne $064a
0659 c6 1b    dec $1b
065b d0 b0    bne $060d
065d 4c 9e fd jmp $fd9e    Ende; zur Jobschleife

0660 a0 00    ldy #$00     Start des Floppyprogramms
0662 b9 e0 06 lda $06e0,y
0665 99 00 02 sta $0200,y  Disknamen übernehmen
0668 c8       iny
0669 cc df 06 cpy $06df
066c 90 f4    bcc $0662
066e ad df 06 lda $06df
0671 8d 74 02 sta $0274    Länge der Zeile setzen
0674 ad de 06 lda $06de
0677 8d 7b 02 sta $027b    Kommaposition setzen
067a a9 00    lda #$00
067c 85 7f    sta $7f      Drive 0 setzen
067e 20 00 c1 jsr $c100    LED am Laufwerk an
0681 ac 7b 02 ldy $027b
0684 b9 00 02 lda $0200,y  ID 1 holen
0687 85 12    sta $12
0689 b9 01 02 lda $0201,y  id 2 holen
068c 85 13    sta $13
068e 20 07 d3 jsr $d307    alle Kanäle schließen
0691 a9 1a    lda #$1a
0693 8d 05 1c sta $1c05    Timer setzen
0696 a9 c0    lda #$c0     BUMP anfordern
0698 85 00    sta $00
069a a5 00    lda $00
069c 30 fc    bmi $069a    auf Ausführung warten
069e ae dc 06 ldx $06dc    erste Tracknummer
06a1 86 0a    stx $0a
06a3 a9 e0    lda #$e0     Track formatieren
06a5 85 02    sta $02
06a7 a5 02    lda $02

06a9 30 fc    bmi $06a7    auf Ende warten
06ab c9 02    cmp #$02     Fehler aufgetreten?
06ad b0 0c    bcs $06bb    verzweige, wenn ja
06af e8       inx
06b0 ec dd 06 cpx $06dd    schon Zieltrack formatiert?
06b3 90 ec    bcc $06a1    weiter, wenn nein
06b5 20 40 ee jsr $ee40    Directory herstellen
06b8 60       rts          Ende
```

Listing 21. Eine neue Formatierroutine.

Ergänzen Sie jetzt Ihre 64'er-Sammlung

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Kennen Sie alle Ausgaben von 64'er? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?

Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel der Ausgaben 01 bis 12/85.

Und so kommen Sie schnell an die noch lieferbaren Ausgaben: Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

Aktuell

Stichwort	Titel	Seite	Ausgabe
Allgemeines	Commodore Gestern Heute Morgen	10	01/85
Computer	Amiga — Der neue Supercomputer	8	09/85
Interview	Interview mit David Crane (Game Designer)	146	06/85
Lernen	Schule braucht Computer (VAM-Computer)	9	06/85
Messen	International Chaos Communication Congress	15	03/85
	Heiße Messe in der Wüste: CES	8	03/85
	Hannover-Messe '85	8	06/85
	Hannover-Messe '85	8	07/85
	Chicago im Zeichen der CES	8	08/85
	Aktuelles von der C'85 in Köln	15	08/85
	Btx Total (Internationale Funkausstellung)	8	10/85
	PCW-Computermesse in London	8	11/85
	Neues von der Commodore-Fachausstellung 1985	8	12/85
Recht	Die neue Abmahnmasche — Vorsicht bei Programmangeboten	8	05/85
	Die Ex-Knacker — wo sind sie geblieben?	27	08/85
	Interview mit Raubkopierern (Section 8)	28	08/85
	Schützer kontra Knacki's	23	08/85
	Raub-Talkshow	12	08/85
	Das Urheberrechtsgesetz und Gedanken zu seiner Anwendung	21	08/85
	Änderung des Urheberrechtsgesetzes	162	09/85

Buchbesprechungen

Stichwort	Titel	Seite	Ausgabe
Anfänger	Goldmann Computer Compact	87	03/85
	Basic-Wegweiser für den C 64	86	05/85
	Alles über den C 64, Sachbuchreihe, Band 1	115	06/85
	Lehrspielzeug Computer: C 64/VC 20	112	11/85
	C 64 Computerhandbuch	171	11/85
	Einführungskurs: Commodore 64	144	12/85
Anwendung	Dienstprogramme VC 20, C 64 und SX	86	05/85
	Spaß an Mathe mit dem Commodore 64	88	07/85
	Mathe für die Oberstufe mit dem C 64	88	07/85
	Mathematische Routinen VC 20, Elektrotechnik/Elektronik	112	11/85
	Commodore 64-Listings, Band 2: Dateiverwaltung, Schule, Hobby	112	11/85
	Das Trainingsbuch zum Datamat	144	12/85
C 128	Bücher zum C 128	22	10/85
DFÜ	Das Mailbox-Jahrbuch: Nutz die Netze	112	11/85
Grafik	Grafik auf dem Commodore 64 (+Fehlert. 9/85)	86	05/85
	Einführung in CAD mit dem Commodore 64	128	06/85
	Grafik & Musik auf dem Commodore 64	88	07/85
	Verschiedene Grafikbücher zum C 64	115	08/85
Programmieren	Von Basic zu Assembler: Das Commodore-Buch, Band 4	115	06/85
	64 Intern	115	06/85
	Das Interface Age System-Handbuch zum C 64	115	06/85
	Das C 64 Buch, Band 5: Simons Basic Leitfaden	144	12/85
	Basicode	144	12/85
	Noch mehr Tips und Tricks zum 64er	144	12/85
Speichern	Das Kassettenbuch zum C 64 und VC 20	87	03/85
	Die Floppy 1541 (M&T)	88	07/85
Spiele	Rombachs C 64 Spielführer	87	03/85
	Commodore 64-Listings, Band 1, Spiele	112	11/85
	35 ausgesuchte Spiele für Ihren Commodore 64	171	1/85

64'er Extra

Stichwort	Titel	Seite	Ausgabe
Prozessor	Befehlssatz des 6502/6510-Prozessors	84	09/85
Grafik	Die Videochip-Register des C 64	92	10/85
Sound	Der SID-Chip, seine Register und Programmierung	92	11/85
Speicher	Die Speicherbelegung des C 64	96	12/85

Abenteuerlösungen

Stichwort	Titel	Seite	Ausgabe
Lösungen	Dallas-Quest Lösung	90	01/85
	Guncho Krill-Enchanter ist gelöst	44	03/85
	Infocom-Geheimnisse gelüftet?	49	05/85
	Des Rätsels Lösung: Amazon	145	06/85
	Activision-Adventures entschleiert (Mindshadow, Tracer Sanction)	36	12/85
	Eureka! — ich hab's!	37	12/85
	Lösungen zu Hitchhiker's Guide und Sorcerer	39	12/85

Spiele-Tests

Stichwort	Titel	Seite	Ausgabe
007	James Bond — A View to a Kill	156	09/85
Abenteuer	Abenteuerpaket 1	48	08/85
	Shadowfire	146	09/85
	The Quest — mit C 64 auf Suche nach Drachen	47	01/85
Action	Hexenküche	50	07/85
	Master of the Lamps	48	07/85
	Rescue on Fractalus	158	10/85
	Stellar 7	49	08/85
Construction Set	Mail Order Monsters	49	08/85
	Racing Destruction Set	50	08/85
Geschicklichkeit	Australopedicus Robustus	50	08/85
	Boulder Dash II	159	10/85
	Crystal Castles	50	07/85
	Gribbly's Day out	148	09/85
	Rock'n Bolt	48	08/85
	Thing on a Spring	159	10/85
	Tom + Zaga	48	01/85
Pseudo-Adventures	Roland's Rat Race	49	08/85
	Fourth Protocol und Frankie g.t.H.	162	11/85
Renner	Die Renner 1985: Meistverkaufte Spiele	34	12/85
Schach	Viermal Schachmatt: Verschiedene Schachprogramme	32	12/85
Simulation	Elite	148	09/85
	Jump Jet	148	09/85
	Super Huey Hubschraubersimulator	49	07/85
Sport	Boxspiele: Frank Bruno's B. + Barry McGuigan Champions. B.	49	12/85
	Handkantenschlag per Joystick: Karateka + Exploding Fist	165	11/85
	Nick Faldo Plays the Open (Golf)	159	10/85
	Rallye Speedway	49	07/85
	Slapshot (Eishockey)	50	07/85
	Summer Games II	146	09/85
	World Series Baseball	49	07/85
Diverses	New York City und Air Support	145	06/85

Hardware-Tips und Bauanleitungen

Stichwort	Titel	Seite	Ausgabe
Audio/Video	Mit 5 Mark zu neuen Dimensionen (Stereoanlage am C 64)	34	05/85
	Ein Monitor ist genug (RGB+Composite an C 128)	16	10/85
C 16	Alte Datasette am C 16	31	04/85
	Alter Joystick am C 16	35	05/85
Eingabegeräte	Der Hexer — Zusatztastatur für den MSE	48	10/85
EPROM	EPROMs im Expansion-Port	46	10/85
	EPROM-Trans — Die Super-Erweiterung	42	10/85
	Das 64'er EPROM-Programmiergerät, Teil 1	44	12/85
Floppy/Datasette	Diskettenlaufwerk 1541 selbst justiert	32	10/85
	Die Datasette streikt nie wieder (Anpassung des Tonkopfs)	34	10/85
IEC-Bus	Auf zu neuen Welten: IEC-Bus im Selbstbau (+Fehlerteufel 10/85)	44	07/85
Joystick	Joystick im Selbstbau	33	03/85
	Dauerfeuer-Adapter	46	08/85
RS232/V.24	Das 30-Mark-Interface (Selbstbau RS232)	29	03/85
	Genau betrachtet: Die RS232/V.24-Schnittstelle	80	05/85
Diverses	Userport-Display	36	05/85
	Reset-Taster für alle Fälle (+Fehlert. 9/85)	130	06/85
	Aus eins mach vier (absturzfreie Betriebssystemumschaltung)	41	07/85

Hardware-Grundlagen

Stichwort	Titel	Seite	Ausgabe
Computer	Was bringt der C 128?	28	11/85
Drucker	Welcher Drucker ist der Richtige? (Grundlagen)	15	05/85
	Hammerwerke — wie funktionieren Typenraddrucker	32	06/85
	Die Alternativen: Thermo-, Tintenstrahldrucker + Plotter	24	07/85
Eingabegeräte	Versteht Sie Ihr Computer? (Wie funktionieren Eingabegeräte)	44	09/85
Floppy	Floppy oder Datasette?	129	06/85
Monitore	Wie funktionieren sie, was ist beim Kauf zu beachten?	16	12/85
	Das Kabel zum Monitor: Welche Normen gibt es?	28	12/85
Peripherie	Grafikeingabegerät: Wie funktionieren sie?	30	08/85

Hardware-Tests

Stichwort	Titel	Seite	Ausgabe
Computer	Generationswechsel: Test C 16	16	01/85
	Erster ausführlicher Test C 128 PC (Teil 1)	16	06/85
	Erster ausführlicher Test C 128, PC (Teil 2)	17	07/85
DFÜ	Marktübersicht Modems & Akustikkoppler	32	07/85
Drucker	Vergleich: Drucker unter 700 Mark (Tests und Marktübersicht)	18	05/85
	Tests und Marktübersicht Typenraddrucker	35	06/85
	Test: Brother EP 44	27	07/85
	Brother TC-600	118	08/85
	Riteman C+	133	09/85
	Panasonic KX-P1091	134	09/85
	Star SG 10C	132	09/85
	Melchers CP-80X — wie hätten Sie's denn gern?	25	10/85
	Geheimtip: Der RFI DP 165	24	10/85
	Epson GX 80 — einer für alle	26	10/85
	MPS 803 — ein Drucker für alle Gelegenheiten?	40	1/85
	Epson JX-80 das vielfarbige Druck-Genie	38	11/85
	Epson FX-85 neue Referenz	42	11/85
	SP 1000 VC — Superstar mit Haken	41	11/85
	Der NEC-P2 — das fernöstliche Wunder	159	12/85
	DMPG9 — eine solide Sache	162	12/85
	Das Doppelleben des Joystick-Ports: 10er-Tastaturen	50	09/85
	Joysticks: Test und Marktübersicht (+Fehlerteufel 12/85)	19	11/85
	Es geht auch anders: Lightpens und Trackballs	22	11/85
EPROMer	Frisch gebrannt ist halb gespeichert (EPROM-Programmiergeräte im Test)	39	07/85
	QuickByte II — das Kraftpaket	14	10/85
Floppy/Datasette	Turbo-Floppies, zweite Generation: Speeddos plus + Prologic DOS	28	10/85
	Das große Rennen: Schnelle Bandlaufwerke	37	10/85
	Professionelle Floppylaufwerke für den C 64 (IEC-Floppies)	30	10/85
	Gut gekauft ist halb gespeichert (Marktübersicht Disketten)	38	10/85
Grafik	Die Videowerkstatt (Digitizer-Test)	32	05/85
	Digitalbilder m.d. C 64: PrintTechnik Digitizer	24	01/85
Interface	Hardware-Interface ganz weich: Test EC 64	23	01/85
	Gute Connections — Übersicht Schnittstellen	21	03/85
	Card/Print +6 — Das Allround-Interface	20	03/85
	Das Wiesemann-Centronics-Interface	18	03/85
	Erst ein IEC-Bus öffnet Tür und Tor (+Fehlert.4/6-85)	24	03/85
Monitore	Marktübersicht: Monochrome Monitore	30	12/85
Musik	Trommelwirbel: Test Digital Drums	45	08/85
	Die Musikhardware zum C 64	17	09/85
Roboter	Roboter selbst gebaut (Fischertechnik)	167	10/85
Scanner	So lernt Ihr Drucker lesen	30	06/85
Speicher	Speichertuning VC 20: Test 64 KByte Karte	26	01/85
Steuern	Flottes Türmchen: MEA-Interface	116	08/85

Kurse

Stichwort	Titel	Seite	Ausgabe
Assembler	Assembler ist keine Alchimie, Teil 5	142	01/85
	Assembler ist keine Alchimie, Teil 7	124	03/85
	Assembler ist keine Alchimie, Teil 9	138	05/85
	Assembler ist keine Alchimie, Teil 10	127	07/85
	Assembler ist keine Alchimie, Teil 11	126	08/85
	Assembler ist keine Alchimie, Teil 12	109	09/85
	Assembler ist keine Alchimie, Teil 13 (Schluß)	143	10/85
C 128	Entdeckungsreise duch den C 128	42	12/85
Effektives Programieren	Müllabfuhr im Computer: Garbage Collection, Teil 1	122	01/85
	Finden mit System, eine neuartige Suchmethode, Teil 3	148	03/85
	Sortieren mit dem Computer, Teil 2	159	05/85
	Sortieren mit dem Computer, Teil 3	124	06/85
	Sortieren mit dem Computer, Teil 4	138	08/85
	Sortieren mit dem Computer, Teil 5	124	09/85
	Sortieren mit dem Computer, Teil 6 (Schluß)	150	12/85
Extern	C 64 extern — Der Weg nach draußen, Teil 1	144	08/85
	C 64 extern — Der Weg nach draußen, Teil 2	122	09/85
	C 64 extern — Der Weg nach draußen, Teil 3 (Schluß)	129	10/85
Floppy	In die Geheimnisse der Floppy eingetaucht, Teil 4	148	01/85
	In die Geheimnise der Floppy eingetaucht, Teil 5	130	03/85
	In die Geheimnise der Floppy eingetaucht, Teil 6	145	05/85
	In die Geheimnisse der Floppy eingetaucht, Teil 7 (Schluß)	116	06/85
	Directory-Manipulationen I	140	06/85
Floppy	Directory-Manipulationen II	163	10/85
Grafik	Hires 3 — 15 neue Basic-Befehle, Teil 2	136	03/85
	Hires 3 — Grafikkurs-Anwendung, Teil 3 (Schluß)	152	08/85
	Sprites ohne Geheimnisse	40	08/85
	Streifzüge durch die Grafikwelt, Teil 1	106	09/85
	Streifzüge durch die Grafikwelt, Teil 2	149	11/85
Logeleien	Logeleien, Teil 1	143	07/85
	Logeleien, Teil 2	136	08/85
	Logeleien, Teil 3 (Schluß)	115	09/85
Musik	Dem Klang auf der Spur, Teil 2	136	01/85
	Dem Klang auf der Spur, Teil 4	131	04/85
	Dem Klang auf der Spur, Teil 5	152	05/85
	Dem Klang auf der Spur, Teil 7	132	07/85
	Dem Klang auf der Spur, Teil 8	133	08/85
	Dem Klang auf der Spur, Teil 9	126	10/85
	Dem Klang auf der Spur, Teil 10 (Schluß)	157	11/85
Speicher	Memory Map mit Wandervorschlägen, Teil 3	126	01/85
	Memory Map mit Wandervorschlägen, Teil 5	144	03/85
	Memory Map mit Wandervorschlägen, Teil 7	120	06/85
	Memory Map mit Wandervorschlägen, Teil 8	140	07/85
	Memory Map mit Wandervorschlägen, Teil 9	129	08/85
	Memory Map mit Wandervorschlägen, Teil 10	112	09/85
	Memory Map mit Wandervorschlägen, Teil 11	133	10/85
	Memory Map mit Wandervorschlägen, Teil 12	145	11/85
	Memory Map mit Wandervorschlägen, Teil 13	146	12/85
Sprachen	Basic ist out — es lebe Forth	43	01/85
VC 20	Der gläserne VC 20, Teil 4	130	01/85
	Der gläserne VC 20, Teil 6 (Schluß)	155	03/85

Software-Tips

Stichwort	Titel	Seite	Ausgabe
C 128	Erste Fragen und Antworten zum C 128	14	09/85
	Fragen und Antworten zum 128er	20	10/85
	Fragen und Antworten zum 128er	40	12/85
Drucker	Der MPS 802 lernt Deutsch	30	05/85
	Centronics-Interface für jeden Bedarf	78	07/85
Textverarbeitung	Software Corner — professionelle Programme richtig eingesetzt (Vizawrite-Tips)	174	12/85
Tips & Tricks	Autoboot beim C 64	86	03/85
	Verbindungsfreundlich (Parallelschnittstelle des VC 20)	91	03/85
	Undefinierte Opcodes des 6502	84	03/85
	Durch POKEs zum Erfolg (Spiele-POKEs)	83	03/85
	Tips und Erweiterungen zu Hi-Eddi und Simons Basic	88	03/85
	Basic-Befehle im Griff	79	05/85
	Durch POKEs zum Erfolg: Spiele-POKEs	78	06/85
	Formatierte Eingabe	148	06/85
	Hi-Text (Text in Hires)	70	08/85
	Verbotene Variablen	66	09/85
	Verschiedene Routinen für Anfänger und Profis (+Fehlerteufel 12/85)	88	11/85
	Der Trick mit dem Joystick (Joystickabfrage)	24	11/85
	Verschiedene Tips für Anfänger und Fortgeschrittene	106	12/85

Software-Grundlagen

Stichwort	Titel	Seite	Ausgabe
Assembler	Assembler? Assembler! (Einführung)	32	01/85
	Assembler-Bedienung leicht gemacht, Teil 1	169	12/85
DFÜ	Der erste Kontakt mit DFÜ	40	06/85
	Die Netze der Post: Btx, Datex-P, Telebox	46	06/85
	DFÜ — Was ist das?	44	06/85
	Mailbox für Anfänger	30	07/85

Listings zum Abtippen

Software-Tests

So machen's andere

Die Ausgaben 2/85 und 4/85 sind bereits vergriffen und nicht mehr lieferbar!

Am besten gleich mitbestellen: Die praktischen 64'er-Sammelboxen



Ein kompletter Jahrgang (12 Ausgaben) paßt in eine der praktischen Sammelboxen! Am besten gleich mitbestellen!

Für alle Leser, die »64'er« regelmäßig kaufen, sammeln oder im Abonnement beziehen, gibt es jetzt ein interessantes Service-Angebot: die 64'er-Sammelbox!

Mit dieser Sammelbox bringen Sie nicht nur Ordnung in Ihre wertvollen Hefte, sondern schaffen sich gleichzeitig ein interessantes und attraktives Nachschlagewerk.

Übrigens: Die Sammelbox ist nicht nur ein praktisches Aufbewahrungsmittel: Sie eignet sich auch hervorragend als Geschenk für Freunde und Bekannte zu vielen Anlässen.

Auch die bisher erschienenen Sonderhefte können Sie jetzt direkt bestellen:

SONDERHEFT 01/84: TIPS & TRICKS
Unentbehrliche Anwendungslistings für C 64 und VC 20.

SONDERHEFT 06/85: AUSGEWÄHLTE SUPER-LISTINGS
Top-Themen aus 64'er bringt eine Auswahl der besten 64'er Programme.

SONDERHEFT 08/85: ASSEMBLER
Assembler-Know-how für Anfänger und Fortgeschrittene.

SONDERHEFT 01/86: PC 128
Komplette Beschreibungen von C 128 und C 128D und passendem Zubehör. Die Unterschiede zum C 64.

SONDERHEFT 02/86: TIPS & TRICKS
Super-Listings, ausführliche Grundlagen und die besten Tips&Tricks und Einzeiler aus 64'er.

SONDERHEFT 03/86: C16, C116, VC20 UND PLUS 4
Umfassende Grundlagen und aktuelle Informationen zu C16, C116, VC20 und Plus 4.

SONDERHEFT 04/86: ABENTEUERSPIELE 2
Auf 160 Seiten alles über das Programmieren von Abenteuerspielen und Super-Listings zum Abtippen.

SONDERHEFT 05/86: C64-GRUNDWISSEN
Für alle Einsteiger umfassende Grundlagen und Hilfestellungen rund um den C64.

SONDERHEFT 06/86: GRAFIK
Grafikprogrammierung des C64, C128 und C128 im C64-Modus. Dreidimensional konstruieren mit »Giga-CAD«.

SONDERHEFT 07/86: PEEKs UND POKEs
Einführungskurs in die wichtigsten Speicherstellen für C64, C16 und C128. Über 30 Seiten Tips&Tricks.

SONDERHEFT 08: PLUS/4 UND C16
Ausführliche Kurse für schnelle Programme auf C16 und Plus/4 in Maschinensprache und Basic mit Grafikbefehlen.

SONDERHEFT 09: FLOPPY & DATEIVERWALTUNG
Die effiziente Datenverwaltung für Einsteiger und Profis.

SONDERHEFT 10: C128II
Entscheidendes Know-how für Anfänger und Fortgeschrittene auf Ihrem Weg zum Profi.

SONDERHEFT 11: GRAFIK, MUSIK, ANWENDUNG
Faszinierende Gestaltungsmöglichkeiten mit Grafik- und Musikprogrammen.

SONDERHEFT 12: ASSEMBLER, PROGRAMMIERSPRACHEN
Erfahren Sie alles über Programmiersprachen und ihre Anwendungsbereiche.

SONDERHEFT 13: HARDWARE
Neue Möglichkeiten für Ihren Computer durch nützliche Hardware-Erweiterungen

SONDERHEFT 14: C16, C116, Plus/4
Super 3D-Grafik-System zum Abtippen

SONDERHEFT 15: TIPS UND TRICKS UND FLOPPY
Alles über Laufwerke und Datasetten. Neue interessante Grundlagen.

SONDERHEFT 16: C64-EINSTEIGER
Ausführliche Grundlagenartikel, komfortable Anwenderprogramme.

SONDERHEFT 17: SPIELE FÜR C64 UND C128
Für jeden etwas! Super-Listings und ausführliche Grundlagen.

SONDERHEFT 18: DRUCKER UND TEXTVERARBEITUNG
Ein Querschnitt durch die gesamte, moderne Drucktechnik und Textverarbeitung.

SONDERHEFT 19: EINSTEIGER
Ausführlicher Basic-Kurs für alle C64-Einsteiger und Super-Spiele zum Abtippen.

SONDERHEFT 20: GRAFIK
Faszinierender Einstieg in die 3D-Welt. Neuer Animations-Editor für »weiche« Bewegungen.

SONDERHEFT 21: ASSEMBLER und BASIC
Giga-Ass: Hypra-Ass hoch 2 / Paradoxon-Basic: 50000 Basic Bytes free

SONDERHEFT 22: C128 III
Farbiges Scrolling im 80-Zeichen-Modus / 8-Sekunden-Kopierprogramm für zwei Laufwerke

SONDERHEFT 23: GRAFIK / ANWENDUNGEN
Paint Magic: ein professionelles Malprogramm.

SONDERHEFT 24: TIPS&TRICKS&TOOLS
Deutsche Spellchecker für Master-Text und Vizawrite

Tragen Sie die Nummer des gewünschten Sonderheftes (z.B. 08/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein.

```
06b9  ea        nop
06ba  ea        nop
06bb  a2 02     ldx #$02
06bd  4c 0a e6  jmp $e60a     Diskstatus ausgeben; Ende

   Computerprogramm zum Disk-Format-System
   (c) 1985 by KOSS
., c200 a2 00     ldx #$00
., c202 20 87 c2 jsr $c287          Titel und erste Frage ausgeben
., c205 a0 00     ldy #$00
., c207 20 cf ff jsr $ffcf          Eingabe holen
., c20a c9 0d     cmp #$0d
., c20c f0 08     beq $c216
., c20e 99 e0 c1 sta $c1e0,y        Namen abspeichern
., c211 c8        iny
., c212 c0 10     cpy #$10          schon 16 Zeichen ?
., c214 90 f1     bcc $c207         weiter, wenn nein
., c216 a9 2c     lda #$2c          Komma hinter den Namen setzen
., c218 99 e0 c1 sta $c1e0,y
., c21b c8        iny
., c21c 8c de c1 sty $c1de
., c21f a2 47     ldx #$47
., c221 20 87 c2 jsr $c287          Frage nach Disk-ID
., c224 a2 00     ldx #$00
., c226 20 cf ff jsr $ffcf          Eingabe abwarten
., c229 c9 0d     cmp #$0d
., c22b f0 09     beq $c236
., c22d 99 e0 c1 sta $c1e0,y        ID ebenfalls abspeichern
., c230 c8        iny
., c231 e8        inx
., c232 e0 02     cpx #$02
., c234 90 f0     bcc $c226
., c236 8c df c1 sty $c1df
., c239 a2 53     ldx #$53
., c23b 20 87 c2 jsr $c287          'FROM TRACK:$' ausgeben
., c23e 20 cf ff jsr $ffcf
., c241 85 fa     sta $fa
., c243 20 cf ff jsr $ffcf
., c246 85 fb     sta $fb
., c248 a9 00     lda #$00
., c24a 85 d0     sta $d0
., c24c a2 62     ldx #$62
., c24e 20 87 c2 jsr $c287          'TO TRACK:$' ausgeben
., c251 20 cf ff jsr $ffcf
., c254 85 fc     sta $fc
., c256 20 cf ff jsr $ffcf
., c259 85 fd     sta $fd
., c25b a9 00     lda #$00
., c25d 85 d0     sta $d0
., c25f a5 fa     lda $fa
., c261 a6 fb     ldx $fb
., c263 20 04 c4 jsr $c404          Umrechnung in HEX-Byte
., c266 8d dc c1 sta $c1dc          Anfangstrack setzen
., c269 a5 fc     lda $fc
., c26b a6 fd     ldx $fd
., c26d 20 04 c4 jsr $c404          Umrechnung in HEX-Byte
., c270 8d dd c1 sta $c1dd          Endetrack setzen
., c273 ee dd c1 inc $c1dd          plus 1 als Vergleichswert
., c276 ea        nop
., c277 ea        nop
., c278 ea        nop
., c279 ea        nop
., c27a ea        nop
., c27b ea        nop
., c27c ea        nop
., c27d ea        nop
., c27e ea        nop
., c27f ea        nop
., c280 ea        nop
., c281 ea        nop
., c282 ea        nop
., c283 ea        nop
., c284 4c 93 c2 jmp $c293          weiter

., c287 bd 4d c3 lda $c34d,x        Ausgabe der Texte
., c28a f0 06     beq $c292
., c28c 20 d2 ff jsr $ffd2
., c28f e8        inx
., c290 d0 f5     bne $c287
., c292 60        rts

., c293 a9 0d     lda #$0d
., c295 20 d2 ff jsr $ffd2
., c298 a9 0d     lda #$0d
., c29a 20 d2 ff jsr $ffd2
., c29d a9 00     lda #$00
., c29f a2 c0     ldx #$c0
., c2a1 85 a7     sta $a7
., c2a3 86 a8     stx $a8
., c2a5 a9 00     lda #$00
., c2a7 a2 05     ldx #$05
., c2a9 85 a9     sta $a9
., c2ab 86 aa     stx $aa
., c2ad a9 08     lda #$08          LISTEN für Gerät Nummer 8
., c2af 20 b1 ff jsr $ffb1
., c2b2 a9 6f     lda #$6f          15; Kommandokanal
., c2b4 20 93 ff jsr $ff93
., c2b7 a9 4d     lda #$4d
., c2b9 20 a8 ff jsr $ffa8          Programm zur Floppy senden
., c2bc a9 2d     lda #$2d
., c2be 20 a8 ff jsr $ffa8
., c2c1 a9 57     lda #$57
., c2c3 20 a8 ff jsr $ffa8
., c2c6 a0 00     ldy #$00
., c2c8 a5 a9     lda $a9
., c2ca 20 a8 ff jsr $ffa8
., c2cd a5 aa     lda $aa
., c2cf 20 a8 ff jsr $ffa8
., c2d2 a9 1e     lda #$1e
., c2d4 20 a8 ff jsr $ffa8
., c2d7 b1 a7     lda ($a7),y
., c2d9 20 a8 ff jsr $ffa8
., c2dc c8        iny
., c2dd c0 1e     cpy #$1e
., c2df 90 f6     bcc $c2d7
., c2e1 20 ae ff jsr $ffae
., c2e4 18        clc
., c2e5 a5 a7     lda $a7
., c2e7 69 1e     adc #$1e
., c2e9 85 a7     sta $a7
., c2eb 90 03     bcc $c2f0
., c2ed e6 a8     inc $a8
., c2ef 18        clc
., c2f0 a5 a9     lda $a9
., c2f2 a6 aa     ldx $aa
., c2f4 69 1e     adc #$1e
., c2f6 85 a9     sta $a9
., c2f8 90 02     bcc $c2fc
., c2fa e6 aa     inc $aa
., c2fc e0 07     cpx #$07
., c2fe 90 ad     bcc $c2ad
., c300 c9 00     cmp #$00
., c302 90 a9     bcc $c2ad
., c304 a9 08     lda #$08          LISTEN für Gerät 8
., c306 20 b1 ff jsr $ffb1
., c309 a9 6f     lda #$6f          15; Kommandokanal
., c30b 20 93 ff jsr $ff93
., c30e a9 4d     lda #$4d
., c310 20 a8 ff jsr $ffa8
., c313 a9 2d     lda #$2d
., c315 20 a8 ff jsr $ffa8          Programm in der Floppy
., c318 a9 45     lda #$45          bei Adresse $0600 starten
., c31a 20 a8 ff jsr $ffa8
., c31d a9 60     lda #$60
., c31f 20 a8 ff jsr $ffa8
., c322 a9 06     lda #$06
., c324 20 a8 ff jsr $ffa8
., c327 20 ae ff jsr $ffae
., c32a a9 00     lda #$00
., c32c 85 90     sta $90
., c32e a9 08     lda #$08          Fehlermeldung holen
., c330 20 b4 ff jsr $ffb4
., c333 a9 6f     lda #$6f
., c335 20 96 ff jsr $ff96
., c338 20 a5 ff jsr $ffa5
., c33b 20 d2 ff jsr $ffd2          und anzeigen
., c33e 24 90     bit $90
., c340 50 f6     bvc $c338
., c342 20 ab ff jsr $ffab
., c345 4c dc c3 jmp $c3dc          Endebehandlung
.: C348 00 00 00 00 00 93 20 20
.: C350 20 20 20 20 20 2A 2A 2A
.: C358 20 44 49 53 4B 2D 46 4F
.: C360 52 4D 41 54 2D 53 59 53
.: C368 54 45 4D 20 2A 2A 2A 0D
.: C370 0D 0D 20 28 43 29 20 31
.: C378 39 38 35 20 42 59 20 4B
.: C380 4F 53 53 20 20 20 0D 0D
.: C388 0D 44 49 53 4B 4E 41 4D
.: C390 45 3A 20 00 0D 0D 44 49
.: C398 53 4B 2D 49 44 3A 20 00
.: C3A0 0D 0D 46 52 4F 4D 20 54
.: C3A8 52 41 43 4B 3A 24 00 0D
.: C3B0 0D 54 4F 20 54 52 41 43
.: C3B8 4B 3A 24 00 0D 0D 41 4E
.: C3C0 4F 54 48 45 52 20 46 4F
.: C3C8 52 4D 41 54 20 28 59 2F
.: C3D0 4E 29 20 3F 20 0D 0D 00
.: C3D8 00 00 00 00 20 29 C4 A2
., c3dc 20 29 c4 jsr $c429          SAVE-Vektor stellen
., c3df a2 6f     ldx #$6f
., c3e1 20 87 c2 jsr $c287          'ANOTHER FORMAT (Y/N) ?' ausgeben
., c3e4 20 e4 ff jsr $ffe4
., c3e7 f0 fb     beq $c3e4
., c3e9 c9 59     cmp #$59          Ende ?
., c3eb d0 03     bne $c3f0         ja: RTS
., c3ed 4c 00 c2 jmp $c200          noch einmal formatieren
., c3f0 60        rts
., c3f1 00        brk
., c3f2 a5 b7     lda $b7           Länge des Filenamens =0?
., c3f4 f0 03     beq $c3f9         ja; dann formatieren
., c3f6 4c ed f5 jmp $f5ed          zur SAVE-Routine
., c3f9 20 00 c2 jsr $c200          formatieren
., c3fc a9 01     lda #$01
., c3fe a2 00     ldx #$00
., c400 a0 00     ldy #$00
., c402 18        clc
., c403 60        rts               Ende

., c404 85 02     sta $02           Umrechnung der Eingabe in ein
., c406 86 03     stx $03           HEX-Byte
., c408 a5 02     lda $02
., c40a c9 41     cmp #$41
., c40c 90 03     bcc $c411
., c40e 18        clc
., c40f 69 09     adc #$09
., c411 29 0f     and #$0f
., c413 0a        asl
., c414 0a        asl
., c415 0a        asl
., c416 0a        asl
., c417 85 02     sta $02
., c419 a5 03     lda $03
., c41b c9 41     cmp #$41
., c41d 90 03     bcc $c422
., c41f 18        clc
., c420 69 09     adc #$09
., c422 29 0f     and #$0f
., c424 05 02     ora $02
., c426 85 02     sta $02
., c428 60        rts

., c429 a9 f2     lda #$f2          SAVE-Vektor herstellen
., c42b 8d 32 03 sta $0332
., c22e a9 c3     lda #$c3          auf Adresse $c3f2 setzen
., c230 8d 33 03 sta $0333
., c233 60        rts
```

Listing 21. (Schluß). Zwischen den Adressen C348 und C3DA liegt eine ASCII-Tabelle.

Es wurde vorhin schon erwähnt, daß die Formatierroutine jeweils über einen Sprungbefehl bei $0600 im RAM der Floppystation aufgerufen wird.

Diese Adresse wird bei jedem neuen Track angesprungen und bietet so die Möglichkeit, Tracks zu erzeugen, die in ihrem Aufbau voneinander abweichen, wenn entsprechende Eingriffe vorgenommen werden.

Diese Möglichkeit eines Eingriffes wollen wir an dieser Stelle aber gar nicht erst weiter diskutieren, da es ziemlich aussichtslos ist, hier ohne dokumentiertes DOS-Listing an die Arbeit zu gehen.

Daß wir kein DOS-Listing besitzen, soll aber noch lange nicht heißen, daß wir nicht in der Lage sind, auf anderem Weg Eingriffe in die Formatierung vorzunehmen. Wenn wir nicht effektiv mit der fest eingebauten Routine zusammenarbeiten können, dann schreiben wir uns eben ein vollständig eigenes Programm, das im RAM der Floppystation abgelegt wird und uns für Abänderungen unendlich viele Möglichkeiten bietet.

Formatierung »selbst gebaut«

Sehen Sie sich einmal Listing 21 an. Es wurde hier ein Formatiersystem entwickelt, das einfacher und schneller arbeitet als die DOS-Routine und trotzdem ein paar zusätzliche Möglichkeiten bietet.

Da das Gesamtprinzip aber fast 100prozentig mit der im DOS eingebauten Routine übereinstimmt, können Sie sich anhand des Source-Code-Listings einmal die »praktische Ausführung« einer Formatierroutine ansehen.

Um Ihnen die Eingabe des Programms zu erleichtern, wurde ein DATA-Lader als Listing 22 beigefügt, wobei wir Ihnen empfehlen möchten, dieses gleich einmal einzutippen.

Das Programm wird nur aktiviert, wenn alle DATAs richtig eingetippt wurden. Haben Sie alles richtig gemacht, so steht nach der Ausführung des Laders ein Maschinenprogramm am Basic-Anfang, dem eine Basic-Zeile beigefügt ist. Das Programm sollten Sie sich jetzt mit SAVE auf eine Diskette speichern und danach mit RUN starten.

Nach einer winzigen Verzögerung erscheint die READY-Meldung und der Cursor wieder. Das Formatierungsprogramm wurde jetzt in den Bereich ab $C000 (49152) geschoben und der SAVE-Vektor abgeändert.

Tippen Sie jetzt einfach den Befehl SAVE – ohne Anführungszeichen und Filenamen – ein und drücken Sie <RETURN>. Es erscheint nun die Startmeldung des Formatprogrammes. Sie können jetzt einen Namen für eine Diskette eingeben (maximal 16 Zeichen werden angenommen). Danach erwartet der Computer eine zweistellige ID. Schließlich, und das ist das Besondere an diesem Programm, können Sie noch den ersten und letzten zu formatierenden Track eingeben. Diese Eingabe muß hexadezimal erfolgen und erlaubt einen Bereich von $01 bis $FF.

Achtung! Wird eine Zahl größer als $29 (41) eingegeben, wird es in der Regel kritisch. Der Kopf ist dann nämlich am oberen Anschlagpunkt angelangt.

Etwas ist noch zu beachten: Ein Nachformatieren einer Spur auf einer gefüllten Diskette ist mit dem Programm ohne Änderung nicht möglich, da das Directory auf jeden Fall neu geschrieben wird. Wird die Diskette nicht vollständig formatiert, so ist darauf zu achten, daß die gleiche ID eingegeben wird, wie sie schon für die übrige Diskette Gültigkeit hat, da es sonst einen »29, DISK ID MISMATCH ERROR« gibt.

Wollen Sie dennoch einen Einzeltrack neu formatieren, ohne das Directory zu zerstören, so können Sie das durch eine einfache Änderung im Floppy-Programm erreichen. Sie gehen in Listing 21 an die Adresse $06B5. Den Befehl JSR $EE40 und das nachfolgende RTS ersetzen Sie durch lauter NOPs.

Eine Änderung des Directory-Track unterbleibt jetzt, sofern Sie die TrackNummern zur Formatierung entsprechend wählen, da dieser Befehl die Routine zum kurzen Formatieren im DOS aufgerufen hätte.

In jedem Fall gilt aber: Bei Formatieren von Einzel-Tracks müssen diese die gleiche ID wie die übrige Diskette erhalten.

Eine weitere Möglichkeit dient der Schonung des Laufwerks. Wenn Sie sich das Floppy-Programm noch einmal betrachten, dann finden Sie bei Adresse $0696 den Befehl an den Diskcontroller, einen BUMP auszuführen. Wenn Sie hier das $C0 durch ein $00 ersetzen, dann unterbleibt dieses Anschlagen des Tonkopfes am Anfang des Formatierens. Diese Maßnahme ist immer dann nützlich, wenn mehrere Disketten hintereinander formatiert werden sollen.

Zur Zeitdauer ist noch zu sagen, daß das Programm für eine Diskette zirka 30 Sekunden benötigt und damit um einiges schneller ist als das Programm im DOS der Floppy 1541. Warum das so ist, sollen wir gleich erfahren.

In meinem Formatierprogramm wurde die Berechnung der Lücke zwischen zwei Sektoren weggelassen. Wir können nämlich davon ausgehen, daß diese Lücken auf jeder Diskette in etwa gleich sind. Aus diesem Grund verwende ich einfach einen Erfahrungswert für die Länge der Lücke, der zusätzlich noch einen Sicherheitsbereich enthält. Diesen Wert sehen Sie in Listing 21 an der Adresse $05DF.

Wenn Sie mit dem Programm Disketten formatieren, werden Sie feststellen, daß die Datensicherheit auch weiterhin voll gewährleistet ist.

Im Gegensatz zu anderen schnellen Formatierprogrammen wurde aber nicht auf ein Verify verzichtet, da das Formatieren die einzige Möglichkeit bietet, defekte Disketten rechtzeitig zu erkennen, ohne daß dabei wichtige Daten verlorengehen. Einmal ganz davon abgesehen, macht das Verifizieren außerdem nur einen sehr kleinen Teil am Geschwindigkeitsverlust aus, so daß die Sicherheit vor einigen Sekunden Zeitgewinn Vorrang haben sollte.

Wollen Sie die Zeit dennoch einmal ohne Verify messen, so »klemmen« sie den Rest der Formatierungsroutine ab $05FD ganz einfach ab, indem Sie an dieser Stelle nach JSR $FE00 ein JMP $FD9E einfügen.

Schneller und besser

Eine weitere Verbesserung gegenüber dem DOS 2.6 der Floppy 1541 hat eigentlich mehr kosmetischen Charakter. Es geht hier um den Leerinhalt von Datenblöcken, nachdem eine Diskette neu formatiert wurde. Den Inhalt werden Sie höchstwahrscheinlich schon kennen: Es steht am Anfang des Datenblocks ein $4B gefolgt von 255 $01-Byte.

Dieser Inhalt ist eigentlich auf einen Fehler im DOS zurückzuführen; er müßte, wie auch bei den großen Commodore-Diskettenlaufwerken aus 256 $00-Byte bestehen.

Im Programm werden alle Sektoren mit dem üblichen Wert $00 gefüllt. Noch ein paar Hinweise zur Benutzung des Formatierprogramms.

Nach RUN wird automatisch der SAVE-Vektor auf den Programmstart der Formatierroutine gestellt. Wird kein Filename angegeben, so erfolgt ein Sprung in das Formatierprogramm. Durch Drücken von <RUN/STOP+RESTORE> läßt sich der SAVE-Vektor wieder richtig »hinbiegen«. Hierzu dürfte jedoch kein Anlaß bestehen, da ansonsten bei fehlendem Filenamen kein Programm gestartet wird.

Mußten Sie dennoch einmal <RESTORE> drücken, so

läßt sich das Formatiersystem mit SYS 49664 ($C200) erneut starten; nach Beendigung wird unter anderem auch der SAVE-Vektor wieder auf das Programm zurückgestellt.

Wollen Sie sich den Disk-Status anzeigen lassen, so tippen Sie SYS49962. Es erscheint danach auch die Frage nach einem weiteren Formatiervorgang. Nach dieser Anzeige wird der SAVE-Vektor ebenfalls wieder hergestellt.

Ich möchte Sie an dieser Stelle auf ein paar Speicherstellen in der Zero-Page der 1541 aufmerksam machen. Wie Sie wissen, werden dort nach einem Reset ein paar Konstanten abgelegt, die vom Benutzer (beliebig) verändert werden können. Mit den Konstanten meine ich zum Beispiel $08 als Kennzeichen eines Blockheaders oder $07 als Kennzeichen eines Datenblocks.

```
10 REM *************************          <137>
20 REM *                       *          <247>
30 REM *  DISK-FORMAT-SYSTEM   *          <052>
40 REM *                       *          <011>
50 REM *   (C) 1985 BY KOSS    *          <091>
60 REM *                       *          <031>
70 REM *************************          <197>
80 DATA 5657,5638,6947,7770,8264,7062,8578
   ,6111,3989,3215,9192,10797             <224>
90 DATA 8104,8232,8308,3524,3180,5204,4577 <144>
100 DATA 0,14,8,10,0,158,32,50,48,54,52,32
    ,32,0,0,0,162,64,160,8,134,2,132,3    <156>
110 DATA 162,0,160,192,134,4,132,5,160,0,1
    62,5,177,2,145,4,200,208,249,230,3    <187>
120 DATA 230,5,202,208,242,120,169,242,141
    ,50,3,169,195,141,51,3,88,96,234,234  <071>
130 DATA 165,10,201,36,144,7,169,18,133,67
    ,76,19,5,32,75,242,133,67,169,0,133   <043>
140 DATA 27,160,0,162,0,165,57,153,0,3,200
    ,200,165,27,153,0,3,200,165,10,153    <201>
150 DATA 0,3,200,165,19,153,0,3,200,165,18
    ,153,0,3,200,169,15,153,0,3,200,153   <254>
160 DATA 0,3,200,169,0,89,250,2,89,251,2,8
    9,252,2,89,253,2,153,249,2,230,27     <218>
170 DATA 165,27,197,67,144,190,169,3,133,4
    9,152,72,138,157,0,7,232,208,250,32   <092>
180 DATA 48,254,104,168,136,32,229,253,32,
    245,253,169,7,133,49,32,233,245,133   <100>
190 DATA 58,32,143,247,169,0,133,50,32,14,
    254,169,255,141,1,28,162,5,80,254     <251>
200 DATA 184,202,208,250,162,10,164,50,80,
    254,184,185,0,3,141,1,28,200,202,208  <125>
210 DATA 243,162,9,80,254,184,169,85,141,1
    ,28,202,208,245,169,255,162,5,80,254  <184>
220 DATA 184,141,1,28,202,208,247,162,187,
    80,254,184,189,0,1,141,1,28,232,208   <122>
230 DATA 244,160,0,80,254,184,177,48,141,1
    ,28,200,208,245,169,85,162,8,80,254   <146>
240 DATA 184,141,1,28,202,208,247,165,50,2
    4,105,10,133,50,198,27,208,149,80     <041>
250 DATA 254,184,80,254,184,32,0,254,169,2
    00,133,31,169,0,133,48,169,3,133,49   <160>
260 DATA 165,67,133,27,32,86,245,162,10,16
    0,0,80,254,184,173,1,28,209,48,208    <123>
270 DATA 14,200,202,208,242,24,165,48,105,
    10,133,48,76,53,6,198,31,208,209,169  <225>
280 DATA 6,76,211,253,32,86,245,160,187,80
    ,254,184,173,1,28,217,0,1,208,231     <087>
290 DATA 200,208,242,162,252,80,254,184,17
    3,1,28,217,0,7,208,215,200,202,208    <125>
300 DATA 241,198,27,208,176,76,158,253,160
    ,0,185,224,6,153,0,2,200,204,223,6    <150>
310 DATA 144,244,173,223,6,141,116,2,173,2
    22,6,141,123,2,169,0,133,127,32,0     <084>
320 DATA 193,172,123,2,185,0,2,133,18,185,
    1,2,133,19,32,7,211,169,26,141,5,28   <206>
330 DATA 169,192,133,0,165,0,48,252,174,22
    0,6,134,10,169,224,133,2,165,2,48     <128>
340 DATA 252,201,2,176,12,232,236,221,6,14
    4,236,32,64,238,96,234,234,162,2,76   <241>
350 DATA 10,230,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0       <204>
360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0        <248>
370 DATA 162,0,32,135,194,160,0,32,207,255
    ,201,13,240,8,153,224,193,200,192     <149>
380 DATA 16,144,241,169,44,153,224,193,200
    ,140,222,193,162,71,32,135,194,162    <233>
390 DATA 0,32,207,255,201,13,240,9,153,224
    ,193,200,232,224,2,144,240,140,223    <206>
400 DATA 193,162,83,32,135,194,32,207,255,
    133,250,32,207,255,133,251,169,0,133  <097>
410 DATA 208,162,98,32,135,194,32,207,255,
    133,252,32,207,255,133,253,169,0,133  <114>
420 DATA 208,165,250,166,251,32,4,196,141,
    220,193,165,252,166,253,32,4,196,141  <125>
430 DATA 221,193,238,221,193,234,234,234,2
    34,234,234,234,234,234,234,234,234    <031>
440 DATA 234,234,76,147,194,189,77,195,240
    ,6,32,210,255,232,208,245,96,169,13   <123>
450 DATA 32,210,255,169,13,32,210,255,169,
    0,162,192,133,167,134,168,169,0,162   <100>
460 DATA 5,133,169,134,170,169,8,32,177,25
    5,169,111,32,147,255,169,77,32,168    <094>
470 DATA 255,169,45,32,168,255,169,87,32,1
    68,255,160,0,165,169,32,168,255,165   <165>
480 DATA 170,32,168,255,169,30,32,168,255,
    177,167,32,168,255,200,192,30,144     <050>
490 DATA 246,32,174,255,24,165,167,105,30,
    133,167,144,3,230,168,24,165,169,166  <204>
500 DATA 170,105,30,133,169,144,2,230,170,
    224,7,144,173,201,0,144,169,169,8     <039>
510 DATA 32,177,255,169,111,32,147,255,169
    ,77,32,168,255,169,45,32,168,255,169  <001>
520 DATA 69,32,168,255,169,96,32,168,255,1
    69,6,32,168,255,32,174,255,169,0,133  <009>
530 DATA 144,169,8,32,180,255,169,111,32,1
    50,255,32,165,255,32,210,255,36,144   <183>
540 DATA 80,246,32,171,255,76,220,195,0,0,
    0,0,0,147,32,32,32,32,32,32,32,42     <048>
550 DATA 42,42,32,68,73,83,75,45,70,79,82,
    77,65,84,45,83,89,83,84,69,77,32,42   <006>
560 DATA 42,42,13,13,13,32,40,67,41,32,49,
    57,56,53,32,66,89,32,75,79,83,83,32   <223>
570 DATA 32,32,13,13,13,68,73,83,75,78,65,
    77,69,58,32,0,13,13,68,73,83,75,45    <197>
580 DATA 73,68,58,32,0,13,13,70,82,79,77,3
    2,84,82,65,67,75,58,36,0,13,13,84     <155>
590 DATA 79,32,84,82,65,67,75,58,36,0,13,1
    3,65,78,79,84,72,69,82,32,70,79,82    <246>
600 DATA 77,65,84,32,40,89,47,78,41,32,63,
    32,13,13,0,0,0,0,32,41,196,162,111    <016>
610 DATA 32,135,194,32,228,255,240,251,201
    ,89,208,3,76,0,194,96,0,165,183,240   <011>
620 DATA 3,76,237,245,32,0,194,169,1,162,0
    ,160,0,24,96,133,2,134,3,165,2,201    <194>
630 DATA 65,144,3,24,105,9,41,15,10,10,10,
    10,133,2,165,3,201,65,144,3,24,105    <176>
640 DATA 9,41,15,5,2,133,2,96,169,242,141,
    50,3,169,195,141,51,3,96,0            <095>
1000 REM                                  <122>
1010 REM **** DATAS INITIALISIEREN        <178>
1020 REM                                  <142>
1030 RESTORE:PRINT:PRINT:PRINT"DATAS WERDE
     N UEBERPRUEFT !!!":PRINT:PRINT       <215>
1040 CLR:DIM P(19):DIM W(19)              <077>
1050 FOR X=0 TO 18:READ P(X):P=P+P(X):NEXT <139>
1060 IF P<>124349 THEN PRINT"PRUEFSUMMENFE
     HLER":PRINT:PRINT:LIST 80-90         <038>
1070 FOR X=0 TO 18:FOR Y=0 TO 59:READ A:W(
     X)=W(X)+A:NEXT Y                     <240>
1080 IF W(X)<>P(X)THEN 1150               <146>
1090 NEXT X                               <032>
1100 PRINT:PRINT"DIE DATAS SIND OK UND WER
     DEN":PRINT:PRINT"ABGESPEICHERT!"     <116>
1110 RESTORE:FOR X=0 TO 18:READ A:NEXT    <166>
1120 FOR X=0 TO 1139:READ A:POKE X+2048,A:
     NEXT                                 <093>
1130 POKE 45,119:POKE 174,119:POKE 46,12:P
     OKE 175,12:CLR                       <006>
1140 PRINT:PRINT"MIT 'SAVE' ABSPEICHERN!":
     PRINT:END                            <097>
1150 REM FEHLERBEHANDLUNG                 <159>
1160 PRINT:PRINT"FEHLER IN DEN DATAS"X*60"
      BIS"X*60+59" !":Z=INT(X*600/17.8)   <161>
1170 PRINT:PRINT:PRINT"DAS ENTSPRICHT IN E
     TWA DEN{14SPACE,DOWN}ZEILEN AB"Z     <073>
1180 END                                  <032>
```

Listing 22. Der DATA-Lader der Formatierroutine

Wie Sie aus der Zero-Page-Belegung entnehmen können, werden diese beiden Werte in den Speicherstellen $47 (Wert 07) und $39 (Wert 08) abgelegt und können nun abgeändert werden. Der neue Wert, den Sie vielleicht in diese Speicherstellen eintragen, sollte sich jedoch im Bereich von $00 bis $0F bewegen, da es sonst Schwierigkeiten beim Lesen geben kann. Die Folge eines Leseversuchs mit normalen Werten, wenn eine Diskette anders formatiert wurde, sind entweder ein »20, READ ERROR« oder ein »22, READ ERROR«.

Der Vorteil dieser Errors ist jedoch die Möglichkeit, den Blöcken auch Inhalte mitzugeben, womit ein sehr wirkungsvoller Kopierschutz konstruiert werden kann.

Zum Lesen oder Beschreiben der Diskette müssen die Werte in den beiden Speicherstellen nur jeweils richtig gestellt werden; dann kann ein ganz normaler Zugriff stattfinden.

Mit Hilfe des Formatierprogramms können Sie jetzt auch noch zusätzlich illegale Spuren beschreiben. Hierbei müssen Sie allerdings, wie vorhin besprochen, auf Job-Schleifenebene arbeiten, um die Begrenzung auf die Spuren 1 bis 35 zu umgehen.

Was ist eine GCR-Codierung?

Vielleicht sind Ihnen bestimmt schon einige Ungereimtheiten aufgefallen, was den Direktzugriff auf die Diskette betrifft. Auch im Abschnitt über das Formatieren waren zum Beispiel im Listing von S-Format einige Sprungbefehle, die nicht erklärt wurden.

Erinnern Sie sich noch an den Abschnitt, der sich das erste Mal mit dem Schreiben von Daten auf die Diskette beschäftigte? Dort wurden unter anderem die SYNC-Markierungen auf der Diskette besprochen, die dem Diskcontroller als Positionsanzeiger dienen.

Es wurde darin erwähnt, daß sich diese SYNC-Markierungen bei der Floppy 1541 aus 5 $FF-Byte zusammensetzen, die hintereinander auf Diskette geschrieben werden. Was ist aber, wenn ein Datenblock geschrieben werden soll, der nur aus $FF-Bytes besteht? Eigentlich müßten dann diese Bytes als SYNC-Markierung wirksam werden und den gesamten Schreib- und Lesebetrieb stören. Wie die Praxis zeigt, tritt dieser Fehler nicht auf. Auch bei mehreren Blöcken aus $FF-Bytes kommt es zu keinen Komplikationen. Bei der Konstruktion der Floppystation hat man sich nämlich eine Codierung der Daten einfallen lassen, die eine Eindeutigkeit der Daten schafft. Die Codierung heißt GCR, was nichts anderes als eine Abkürzung der englischen Wörter »Group Code Recording« ist.

Es stellt sich jetzt die Frage, was bei der GCR-Codierung passiert, damit eine Verwechslung zwischen SYNC- und Daten-Bytes unmöglich wird. Zur Beantwortung dieser Frage muß ein wenig intensiver auf das Lesen und Schreiben der Floppystation eingegangen werden.

Was macht die GCR-Codierung?

Das Lesen von Bytes durch den Lesekopf steuert ein Timer des Diskcontrollers. Auf der Diskette selbst wird jedes 1-Bit physikalisch durch einen Wechsel der Magnetisierungsrichtung dargestellt und ein 0-Bit durch gleichbleibende Richtung der Magnetisierung. Bild 8 zeigt, was gemeint ist. Soll ein Byte von Diskette gelesen werden, so wartet der Diskcontroller einfach die Zeitspanne ab, die zum Lesen von 8 Bit erforderlich ist. Innerhalb dieser Zeit liest der Schreib-/Lesekopf eine gewisse Folge von Magnetisierungs- und Nicht-Magnetisierungswechseln.

Dazu ein Beispiel: Auf der Diskette steht ein Byte mit dem In- $55. $55 wird binär durch die Kombination %01010101 dargestellt. Der Tonkopf stellt also während der Lesezeit die folgenden Magnetisierungswechsel fest:

Magnetisierung wechselt nicht, wechselt, wechselt nicht, wechselt, wechselt nicht, wechselt, wechselt nicht, wechselt.

```
N/N        S/S N/N S/S     N/N S/S       N/N      S/S
 *          *   *   *       *   *         *        *

 1  0  0  1  1  1  0  1  1  0  0  1  0  1
```

* = Magnetisierungswechsel

Bild 8. Die Aufzeichnung von Daten auf Diskette (schematisch)

Das Erkennen eines Bits geschieht dabei völlig zeitgesteuert. Der Diskcontroller »weiß«, daß er zum Lesen eines Bits eine bestimmte Zeit warten muß. Danach gilt das Bit als gelesen, und es wird eine »1« oder eine »0« bereitgestellt, je nachdem, ob ein Magnetisierungswechsel stattgefunden hat oder nicht.

Praktisch könnte man das folgendermaßen beschreiben: Sie machen mit einem Freund eine Zeit von 10 Sekunden

Hexadezimal	Binär	GCR
$0	0000	01010
$1	0001	01011
$2	0010	10010
$3	0011	10011
$4	0100	01110
$5	0101	01111
$6	0110	10110
$7	0111	10111
$8	1000	01001
$9	1001	11001
$A	1010	11010
$B	1011	11011
$C	1100	01101
$D	1101	11101
$E	1110	11110
$F	1111	10101

Tabelle 17. Umrechnungstabelle für Binär-GCR-Umwandlung

aus. Er hat dann die Aufgabe, innerhalb dieser 10 Sekunden entweder zu pfeifen oder nicht. Danach warten Sie diese 10 Sekunden ab. Hat er während dieser Zeit gepfiffen, dann entspricht das einem Magnetisierungswechsel. Hat er innerhalb der 10 Sekunden nicht gepfiffen, bedeutet das ein »0«-Bit, also keinen Magnetisierungswechsel. Da eine Diskette im Laufwerk nicht absolut gleichmäßig gedreht werden kann, also Drehzahlschwankungen unterliegt, muß noch für eine Kompensation der mechanischen Fehler gesorgt werden. Dazu wird der Timer, der die abzuwartende Zeit für jedes Bit bestimmt, bei jedem Magnetisierungswechsel neu getriggert (gestellt). Ein »1«-Bit hat also neben seinem Informationsgehalt noch die wichtige Aufgabe, Laufwerksschwankungen auszugleichen, um Lesefehler zu verhindern. Aus diesem Grund darf es zum Beipiel nicht

passieren, daß mehrere $00-Bytes hintereinander auf der Diskette stehen, da sonst zu lange keine Laufwerkskontrolle mehr stattfinden könnte.

Aber auch zu viele »1«-Bits sind nicht gestattet, da mehr als acht »1«-Bits ein SYNC-Signal auslösen.

Aus den genannten Gründen werden alle Daten, die auf die Diskette geschrieben werden, vorher GCR-codiert. Mit dieser Codierung wird ausgeschlossen, daß mehr als acht »1«-Bit und mehr als zwei »0«-Bit direkt hintereinander auf die Diskette geschrieben werden und so die Schreib- und Lese-Elektronik durcheinanderbringen.

Einzig und allein die SYNC-Markierungen (mehr als acht »1«-Bit) werden vom DOS (Disk Operating System, Controller) uncodiert auf die Diskette geschrieben.

Es gibt zwei Schreibarten

Man kann also zwischen zwei Schreibarten auf Diskette unterscheiden:

1) Schreiben von Markierungen.
Hier werden fünf $FF-Byte direkt hintereinander auf die Diskette geschrieben, um eine SYNC-Markierung zu bilden, die der Orientierung dient.

2) Schreiben von Daten.
In diesem Modus werden Byte-Inhalte codiert, um sich von den Markierungen zu unterscheiden.

Sehen Sie sich jetzt einmal Tabelle 17 an, die Umwandlungstabelle für die Konvertierung Binär nach GCR und umgekehrt.

Wie Sie erkennen können, handelt es sich beim GCR-Code um einen 5-Bit-Code. Jedes 4-Bit-Nibble, das Sie umwandeln, wird zu einem 5-Bit-GCR-Nibble. Ein Byte, das vorher aus 8 Bit bestand, wird also durch die Codierung 10 Bit lang. Allgemein nimmt die Länge der codierten Daten um den Faktor 5/4 zu. Deshalb ist die Handhabung der GCR-Bytes nicht ganz einfach. Wandeln Sie doch einmal zwei Byte in den GCR-Code um. Als Ergebnis erhalten Sie »zweieinhalb« Byte, die sicherlich schwer zu behandeln sind.

Bei der GCR-Codierung geht man aus diesem Grund einen ganz einfachen Weg, um keine Formatprobleme zu bekommen: es werden jeweils immer 4 Byte gleichzeitig umgewandelt. Als Ergebnis erhält man 5 vollständige Byte, die ohne Probleme weiterverarbeitet werden können.

Lassen Sie mich das einmal an einem Beipiel erläutern:

Nehmen wir einmal an, wir hätten vier Byte mit dem Wert $FF. Eine Kombination also, die nicht direkt auf die Diskette geschrieben werden darf.

Wir wandeln diese vier Hex-Byte nun in die entsprechenden fünf GCR-Byte um, indem wir in Tabelle 17 nachsehen, was die entsprechenden GCR-Äquivalente dieser Bytes sind. Wir kommen zu folgendem Ergebnis:

HEX	BINÄR	GCR-Code
$FF	1111 1111	10101 10101
$FF	1111 1111	10101 10101
$FF	1111 1111	10101 10101
$FF	1111 1111	10101 10101

Die binär dargestellten GCR-Werte müssen wir jetzt nur noch zu fünf Byte zusammenfassen, um auf folgendes Ergebnis zu kommen:

1010 + 1101 = AD
(1010 1 + 101 01)
0110 + 1011 = 6B
0101 + 1010 = 5A
1101 + 0110 = D6
1011 + 0101 = B5

Vier $FF-Byte werden also bei der GCR-Codierung in die fünf Byte $AD, $6B, $5A, $D6 und $B5 umgewandelt. Sie können sich jetzt leicht davon überzeugen, daß diese fünf Byte für den Diskcontroller absolut ungefährlich sind, und daß sie die vorgeschriebenen Normen (nicht mehr als zwei »0«-Byte und nicht mehr als acht »1«-Byte) erfüllen.

Um Ihnen die Umwandlung der Bytes zu erleichtern, habe ich diesem Kurs zwei Programmlistings beigefügt. Listing 23 enthält ein Programm, das Ihnen vier Hex-Byte in fünf GCR-Byte umwandelt. In Listing 24 sehen Sie ein Programm abgedruckt, das die GCR-Codierung wieder rückgängig macht. Hier werden fünf GCR-Byte in vier Hex-Bytes zurückverwandelt, wobei Sie mit unerlaubten Bitkombinationen vorsichtig sein sollten. Kann ein Byte nicht zurückverwandelt werden, so haben Sie eine unerlaubte GCR-Bitkombination, die sich im Ergebnis dadurch äußert, daß entsprechende Nibbles fehlen. Sie erhalten dann unter Umständen nur »halbe« Bytes.

Die Floppystation hält übrigens für diesen Fall eine Fehlermeldung bereit, einen »24, READ ERROR«.

Im DOS existieren übrigens die folgenden Routinen zur Konvertierung:

$F6D0: Dieses Programm holt vier Hex-Byte aus den

```
10 REM PROGRAMM ZUR KONVERTIERUNG                 <242>
20 REM VON VIER HEXBYTES IN DIE                   <161>
30 REM FUENF ENTSPRECHENDEN                       <055>
40 REM GCR-AEQUIVALENTE                           <068>
50 REM                                            <193>
60 REM                                            <203>
70 REM                                            <213>
80 REM (W) 1985 BY KARSTEN SCHRAMM                <028>
90 REM                                            <233>
100 A$="0123456789ABCDEF":DIM G$(15):E$=""        <220>
110 G$(0)="01010"                                 <066>
120 G$(1)="01011"                                 <078>
130 G$(2)="10010"                                 <088>
140 G$(3)="10011"                                 <100>
150 G$(4)="01110"                                 <111>
160 G$(5)="01111"                                 <123>
170 G$(6)="10110"                                 <133>
180 G$(7)="10111"                                 <145>
190 G$(8)="01001"                                 <154>
200 G$(9)="11001"                                 <166>
210 G$(10)="11010"                                <216>
220 G$(11)="11011"                                <228>
230 G$(12)="01101"                                <238>
240 G$(13)="11101"                                <250>
250 G$(14)="11110"                                <005>
260 G$(15)="10101"                                <016>
270 PRINT"{CLR}HEX - GCR - KONVERTIERUNG":
    PRINT                                         <096>
280 PRINT:PRINT"GEBEN SIE JETZT 4 HEXBYTES
    EIN":PRINT                                    <235>
290 PRINT"Z.B. ED 34 27 58":INPUT"{2DOWN}"
    ;H$:GC$=""                                    <104>
300 GOSUB 470:FOR X=1 TO 4                        <035>
310 H1$=MID$(H$,X*2-1,1):H2$=MID$(H$,X*2,1
    )                                             <120>
320 H1=VAL(H1$):H2=VAL(H2$)                       <057>
330 IF H1=0 AND H1$<>"0"THEN H1=ASC(H1$)-5
    5                                             <240>
340 IF H2=0 AND H2$<>"0"THEN H2=ASC(H2$)-5
    5                                             <254>
350 GC$=GC$+G$(H1)+G$(H2)                         <044>
360 NEXT X                                        <067>
370 FOR X=1 TO 10                                 <052>
380 B=0:B$=MID$(GC$,X*4-3,4)                      <094>
390 FOR Y=0 TO 3                                  <026>
400 IF MID$(B$,Y+1,1)="1"THEN B=B+2↑(3-Y)         <112>
410 NEXT Y                                        <118>
420 E$=E$+MID$(A$,B+1,1)                          <249>
430 IF X/2=INT(X/2)THEN E$=E$+" "                 <121>
440 NEXT X                                        <147>
450 PRINT:PRINT:PRINT"GCR: ";E$                   <000>
460 END                                           <077>
470 X$="":FOR X=1 TO LEN(H$)                      <099>
480 IF MID$(H$,X,1)<>" "THEN X$=X$+MID$(H$
    ,X,1)                                         <223>
490 NEXT                                          <109>
500 H$=X$:RETURN                                  <087>
```

Listing 23. Umwandlung von Daten in GCR-Bytes

Speicherstellen $52 bis $55 und wandelt diese Bytes in die fünf entsprechenden GCR-Werte um. Diese fünf Byte werden anschließend im Puffer der Adresse $30/31 (L,H) mit dem Pufferzeiger in $34 abgelegt.

Pufferadresse und Pufferzeiger müssen dabei vor Aufruf dieser Routine übergeben werden.

$F78F: Diese Routine wandelt einen gesamten Puffer, dessen Adresse in $30/31 (L,H) stehen muß, in GCR-Werte um und speichert diese in den Ausweichpuffer sowie den ursprünglichen Puffer zurück. Der Pufferinhalt vergrößert sich durch diese Umwandlung von 256 auf 324 Bytes.

$F7E6: Diese Routine wandelt fünf GCR-Byte aus einem Puffer (dessen Adresse in $30/31 (L,H) und dessen Pufferzeiger in $34 steht) in vier Hex-Byte zurück, wobei diese dann in der Zero-Page von $52 bis $55 gespeichert werden.

$F8E0: Diese Routine decodiert einen gesamten GCR-Pufferinhalt in die ursprüngliche Form und legt diese 256 Byte dann im Puffer mit der Adresse $30/31 (L,H) ab. Die vorherigen 324 GCR-Byte müssen im gleichen Puffer und im Ausweichpuffer ($01BB bis $01FF) stehen.

```
10 REM PROGRAMM ZUR KONVERTIERUNG                  <242>
20 REM VON FUENF GCR-BYTES IN DIE                  <003>
30 REM VIER ENTSPRECHENDEN                         <249>
40 REM HEX-AEQUIVALENTE                            <077>
50 REM                                             <193>
60 REM                                             <203>
70 REM                                             <213>
80 REM (W) 1985 BY KARSTEN SCHRAMM                 <028>
90 REM                                             <233>
100 A$="0123456789ABCDEF":DIM G$(15):E$=""         <220>
110 G$(0)="01010"                                  <066>
120 G$(1)="01011"                                  <078>
130 G$(2)="10010"                                  <088>
140 G$(3)="10011"                                  <100>
150 G$(4)="01110"                                  <111>
160 G$(5)="01111"                                  <123>
170 G$(6)="10110"                                  <133>
180 G$(7)="10111"                                  <145>
190 G$(8)="01001"                                  <154>
200 G$(9)="11001"                                  <166>
210 G$(10)="11010"                                 <216>
220 G$(11)="11011"                                 <228>
230 G$(12)="01101"                                 <238>
240 G$(13)="11101"                                 <250>
250 G$(14)="11110"                                 <005>
260 G$(15)="10101"                                 <016>
270 PRINT"{CLR}GCR - HEX - KONVERTIERUNG":
    PRINT                                          <096>
280 PRINT:PRINT"GEBEN SIE JETZT 5 GCR-BYTE
    S EIN":PRINT                                   <016>
290 INPUT"{2DOWN}";H$:GC$=""                       <147>
300 X$="":FOR X=1 TO LEN(H$)                       <185>
310 IF MID$(H$,X,1)<>" "THEN X$=X$+MID$(H$
    ,X,1)                                          <053>
320 NEXT                                           <195>
330 H$=X$                                          <229>
340 FOR X=1 TO 10                                  <022>
350 X$=MID$(H$,X,1)                                <245>
360 XX=VAL(X$):IF XX=0 AND X$<>"0"THEN XX=
    ASC(X$)-55                                     <104>
370 FOR Y=0 TO 3                                   <006>
380 YY=INT(XX/2↑(3-Y)):XX=XX-YY*2↑(3-Y)            <105>
390 IF YY THEN GC$=GC$+"1":GOTO 410                <240>
400 GC$=GC$+"0"                                    <189>
410 NEXT Y,X                                       <250>
420 HC$="":FOR X=1 TO 8                            <028>
430 X$=MID$(GC$,X*5-4,5)                           <075>
440 FOR Y=0 TO 15                                  <127>
450 IF X$<>G$(Y)THEN NEXT Y                        <197>
460 :                                              <007>
470 HC$=HC$+MID$(A$,Y+1,1)                         <206>
480 IF INT(X/2)=X/2 THEN HC$=HC$+" "               <055>
490 NEXT X                                         <197>
500 PRINT:PRINT:PRINT"HEX: ";HC$                   <129>
```

Listing 24. Umwandlung von GCR- in Daten-Bytes

Die Anwendungen dieser Routinen sind äußerst vielfältig. So können Sie diese Programme zum Beispiel für einen Diskmonitor verwenden, in dem man zwischen der Anzeige von GCR-Bytes und der Anzeige von normalen Hex-Bytes hin- und herschalten kann. Die einzigen Änderungen, die Sie dazu machen müssen, bestehen in der Umrechnung der Adressen für die Speicherbereiche im Computer und der Angabe neuer Parameter als Puffer- und Zero-Page-Bereiche. Ihrer Phantasie, was die Möglichkeiten des Monitors angeht, sind außer dem Speicherplatz im Computer keine Grenzen gesetzt.

Bis zu 365 Byte in einem Block

Durch die Verwendung der GCR-Codierung ergeben sich noch Konsequenzen. Wie sieht es beispielsweise in den Puffern der Floppystation aus, wenn ein Puffer mit einem vollständigen Datenblock (also 256 Byte) gefüllt wurde und dieser aufgezeichnet werden soll? Für dieses Problem hat der Controller einen speziellen Ausweichpuffer. Der Puffer hat eine Größe von 68 Byte und befindet sich im Bereich von $01BB bis $01FF.

Wird nun ein Datenblock in Puffer 1 ($0400 bis $04FF) codiert, so werden die ersten 68 GCR-Byte in den Ausweichpuffer übernommen. Die restlichen Bytes stehen in Puffer 1.

Aus den 256 Byte an Information macht das DOS durch die Konvertierung also 324 Byte, die einen gesamten Datenblock darstellen (inklusive Prüfsumme). Natürlich werden auch die Parameter im Datenblock-Header (ID, Track, Sektor, Prüfsumme und Kennzeichen) vor dem Schreiben auf die Diskette in GCR-Bytes umgewandelt, wobei der Blockheader dann mit den zwei Lücken-Byte auf eine Länge von zehn GCR-Byte anwächst, da der Header aus ursprünglich acht Hex-Werten besteht.

Zusammenfassend besteht ein Sektor auf der Diskette aus den fünf Byte der ersten SYNC-Markierung; danach folgen die zehn Byte des Blockheaders. Vor der SYNC-Markierung des Datenblocks folgen jedoch noch neun $55-Byte, die der GCR-Norm entsprechen und direkt auf die Diskette geschrieben werden. Sie dienen als Pufferlücke, in der dem Diskcontroller Zeit bleibt, zwischen Schreiben und Lesen umzuschalten.

Nach den fünf Byte der SYNC-Markierung folgen die 324 Byte des Datenblocks inklusive dessen Prüfsumme und anschließend noch die Lücke zwischen zwei Sektoren, die erfahrungsgemäß zwischen acht und zwölf Byte lang ist. Wie Sie sehen, hat also so ein Sektor auf der Diskette die stattliche Länge von 361 bis 365 Byte.

Jetzt werden Ihnen bestimmt auch ein paar zweifelhafte JSR-Befehle im vorigen Abschnitt klar: bei dem Formatiersystem wird einmal ein Befehl JSR $FE30 und an anderer Stelle ein Befehl JSR $F78F ausgeführt. Diese Adressen sind die Einsprünge der Codierroutinen.

Vielleicht kommt Ihnen auch noch einmal die Herstellung eines Killertracks in Erinnerung. Hier wird ein gesamter Track direkt mit $FF-Bytes vollgeschrieben und stellt so eine »Riesen-SYNC-Markierung« dar. Da eine solche Bitfolge jedoch unzulässig ist, kommt die Lese-und Schreibelektronik der Floppystation völlig aus dem Konzept; der Controller »stürzt ab«.

Wenn Sie noch mehr über Ihre Floppy 1541, über schnelle Kopierprogramme und Kopierschutz-Methoden erfahren, so sollten Sie den nächsten Artikeln Ihre Aufmerksamkeit widmen. Dort erfahren Sie weitere »Feinheiten« der Floppy-Programmierung. Auch die Laufwerke 1570/71, die weit leistungsfähiger als die 1541 sind, werden dort genau beleuchtet. (Karsten Schramm/sk)

Burst-Modus – was ist das?

Die Floppy 1570/71 kennt neben dem Standard-Befehlssatz, der dem der 1541 entspricht, auch noch die Befehle für den Betrieb am schnellen seriellen Bus. Diese sogenannten User-0-Befehle sollen uns jetzt interessieren.

Wenn man sich die Unterschiede zwischen der Floppy 1571 und der Floppy 1541 vor Augen hält, dann sticht besonders ein Merkmal der 1571 (Bild 1) ins Auge: Sie ist schneller als die 1541. Bei der Arbeit mit dem C 128, insbesondere dem Laden und Speichern von Programmen oder Daten, gewöhnt man sich sehr schnell an diese angenehme Eigenschaft der Floppy 1571. Sollen jedoch eigene Programme entwickelt werden, stellt sich die Frage: Und wie kann ich nun die hohe Geschwindigkeit auch für mich ausnutzen? Die 1571 befindet sich nämlich standardmäßig auf der niedrigsten Geschwindigkeitsstufe (wie die 1541) und muß erst »von Hand« auf den schnellen seriellen Bus umgestellt werden.

Ein wichtiges Werkzeug: der »Burstmon«

Befindet sich die Floppystation in der Betriebsart des schnellen seriellen Bus, so sind die normalen Übertragungsroutinen und auch Befehle des Laufwerks, die wir schon von der 1541 kennen, wirkungslos. Es stehen dem Anwender jetzt neue Befehle zur Verfügung. Diese neuen Befehle beginnen alle mit der Zeichenfolge »U0«, weshalb wir sie »User-0« oder »U0-Befehle« nennen wollen; zusätzlich sind sie so aufgebaut, daß sich mit ihrer Hilfe sämtliche Diskettenoperationen durchführen lassen. Wir werden noch jeden einzelnen dieser Befehle ausführlich besprechen. Vorher wollen wir Ihnen jedoch noch ein mächtiges Werkzeug an die Hand geben, das Ihnen den direkten Zugriff auf alle U0-Befehle der 1571 gestattet. Es handelt sich um den »Burstmon« (Listing 1). Zusätzlich zum Aufrufen sämtlicher Burst-Befehle, können Sie auch Disketteninhalte betrachten, ändern und wieder auf die Diskette zurückschreiben. Sie bekommen also zusätzlich einen leistungsfähigen Diskettenmonitor. Der Burstmon wurde übrigens dem Buch »Die Floppy 1570/1571« entnommen, das im Markt & Technik Verlag unter der Bestellnummer MT 90185 zu erhalten ist.

Wenn wir jetzt die einzelnen Befehle des Burstmon besprechen, so kann es sein, daß Sie das eine oder andere Kommando noch nicht kennen. Wir besprechen diese Befehle dann, wie schon erwähnt, im Anschluß daran sehr ausführlich.

Für die Arbeit mit dem Burstmon benötigen Sie einen C 128 mit RGB-Monitor im 80-Zeichen-Modus. Tippen Sie das Programm mit dem »Checksummer 128« (Seite 158) ein, speichern es auf eine Diskette und starten es anschließend mit RUN.

Alle Formate lesbar

Da die U0-Befehle der 1571 formatunabhängig arbeiten, können Sie sowohl Disketten im C 128-Format als auch andere, MFM-fremdformatierte Disketten mit dem Burstmon bearbeiten. Eine Anpassung erfolgt in der 1571 automatisch.

Nach dem Start des Burstmon mit RUN wird die Bildschirmmaske aufgebaut, die für die gesamte Arbeit mit dem Programm erhalten bleibt. Dabei sind drei verschiedene Fenster auf dem Bildschirm zu erkennen:
Oben: Fenster mit der Anzeige des Controller-Burst-Status
Mitte: Arbeitsfenster (im Augenblick leer)
Unten: Kommandofenster für die Befehlseingabe

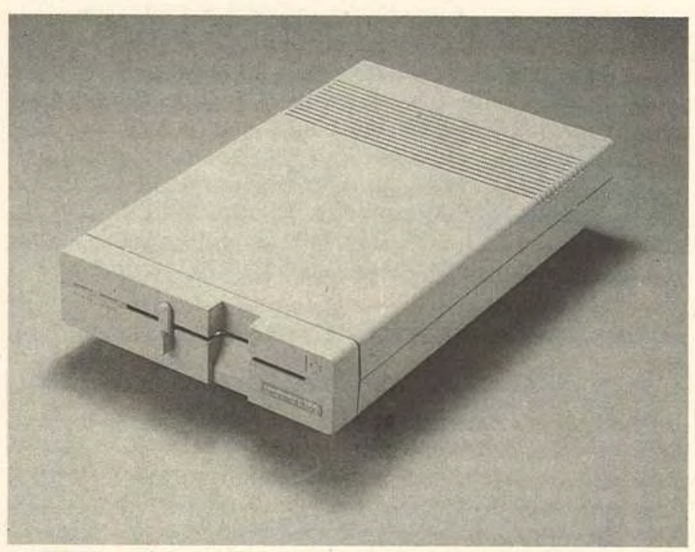


Bild 1. Die Floppy 1571, ein Laufwerk, in dem ungeahnte Möglichkeiten stecken.

Das obere Fenster zeigt Ihnen den aktuellen Controller-Burst-Status an. Dabei handelt es sich um eine Umsetzung des Burst-Status-Bytes in eine Klartext-Meldung. Weitere zusätzlich angezeigte Parameter sind:

Track –	Spurnummer, die gerade bearbeitet wird
Sector –	Sektornummer, die gerade bearbeitet wird
Format –	Format der letzten bearbeiteten Diskette
Sector width –	Größe der physikalischen Sektoren der aktuellen Spur
Capacity –	mögliche Sektoranzahl auf der aktuellen Spur

Ist die Textfarbe der Meldungen im oberen Bildschirmfenster Rot, so ist ein Fehler aufgetreten. Wird vom Burstmon eine Meldung ausgegeben, die eine Bestätigung oder eine Eingabe verlangt, so wechselt die Schriftfarbe auf Gelb. Normalerweise erscheint die Schrift jedoch in der Farbe Cyan.

Burst-Kommandos

Das untere Bildschirmfenster ist, wie schon erwähnt, das Fenster für die Eingabe von Kommandos an den Burstmon. Es erfolgt aber zusätzlich die Anzeige der Statusmeldung der Floppystation. Diese ist nicht mit dem Controller-Burst-Status zu verwechseln, auch wenn sich die Meldungen manchmal ähneln.

Im Augenblick wartet der Monitor mit einem blinkenden Cursor auf Ihre Eingabe im unteren Fenster. Bis auf das Kommando »X« für »Exit«, das Verlassen des Monitors, sind alle Befehle oder Befehlskomplexe auf Funktionstasten gelegt (Tabelle 1). Sie können aber auch direkt eingegeben werden, wobei der Anfangsbuchstabe des betreffenden Kommandos genügt.
Der »Burst-Read«- und »Burst-Write«-Befehl dient jeweils zum Lesen und Schreiben eines Sektors. Die beiden Kom-

mandos erwarten zusätzlich die Angabe der Spur- und Sektornummer sowie die Diskettenseite (0 oder 1). Anschließend wird der Befehl sofort ausgeführt. Haben Sie einen Sektor eingelesen, so gelangen Sie im Anschluß daran direkt in den Edit-Modus des Burstmon; dazu gleich mehr.

Eine kleine Bemerkung sei an dieser Stelle angebracht: Der Burstmon erwartet grundsätzlich eine Quittierung durch Tastendruck, wenn ein Kommando ausgeführt wurde, bei dem eine neue Statusmeldung im oberen oder unteren Fenster generiert wurde.

Der Directory-Befehl zeigt Ihnen das aktuelle Directory einer eingelegten Diskette im 1571- oder 1541-Format an. Maßgebend für das angeschlossene Laufwerk ist die eingestellte Gerätenummer. Der Burstmon kann nämlich sowohl ein Laufwerk mit der Nummer 8 als auch mit der Nummer 9 bearbeiten.

Der Befehl »Disk-Status« zeigt den aktuellen Status der 1571 - nicht den Controller-Burst-Status - im unteren Bildschirmfenster an. Diese Funktion wird auch durch das Eintippen des »Klammeraffen« (@) aufgerufen.

Das Inquire-Disk-Kommando initialisiert eine neueingelegte Diskette für die weitere Bearbeitung und muß immer als erstes Kommando des Burstmon nach einem Diskettenwechsel angegeben werden.

Das Query-Disk-Format-Kommando kann als Ersatz für Inquire-Disk dienen, braucht jedoch für seine Ausführung etwas mehr Zeit. Die Floppystation untersucht nämlich eine unbekannte Diskette auf ihr Format, wobei nahezu sämtliche Format-Parameter »herausgefunden« und angezeigt werden. Die Anzahl der Parameter ist dabei bei MFM-Disketten größer, da es hier viel mehr Kombinationen als bei den Disketten im Commdore-Format gibt.

Fremdformate im Griff

Vergessen Sie nach einem Diskettenwechsel die Angabe eines Inquire-Disk oder Query-Disk-Format-Kommandos, so kann es passieren, daß sich die 1571 »aufhängt«, wenn ein Formatwechsel stattgefunden hat. Die Folge ist, daß der Burstmon »ewig« auf eine Rückmeldung der Floppystation wartet, die natürlich nicht kommt. Hier hilft nur das Aus- und Wiedereinschalten der 1571 und ein <RUN/STOP RESTORE> und erneutes RUN beim C 128. Das ist kein Fehler des Burstmon, sondern liegt an einer Unzulänglichkeit des DOS der 1571, das auf einen solchen »Spezialfall« nicht vorbereitet ist.

Bei der Anwahl des User-0-Befehlssatzes wird ein Zusatzmenü eingeblendet, das sämtliche U0-Befehle enthält. Wird mit der Taste <9> der Utility-Befehl aufgerufen, so erscheint ein weiteres Menü, das eine Auswahl unter den verschiedenen Utility-Befehlen gestattet. Sie gestatten die Einstellung verschiedener System-Parameter der 1571. Da die Erläuterung im Handbuch zur 1570/71 ausführlich genug ist, wollen wir im Rahmen dieses Artikels auf eine Erklärung der Utility-Befehle verzichten.

Die Kommandos in der Reihenfolge der Funktionstasten:	
f1	BURST-READ; Lesen eines Sektors von der Diskette
f2	BURST-WRITE; Schreiben eines Sektors auf eine Diskette
f3	DIRECTORY; Anzeige des Directory für das angewählte Laufwerk
f4	DISK-STATUS anzeigen
f5	INQUIRE DISK; Initialisieren einer Diskette
f6	QUERY DISK; Analysieren des Diskformats auf einer Spur
f7	USER-0-Befehlskomplex
f8	EDIT; Aufrufen des Editors

Tabelle 1. Die Funktionstastenbelegung des Burstmon

Mächtiger Edit-Modus

Der Edit-Befehl wurde schon erwähnt. Er wird automatisch nach dem Lesen eines Sektors aufgerufen, kann jedoch auch von Hand gestartet werden. Der Computer schaltet nun die Anzeige im mittleren Fenster (Arbeitsfenster) auf einen Hex-Dump des aktuellen Sektorinhalts um, und ein Cursor erscheint. Nun können sämtliche Bytes in dem Sektor geändert werden. Es reicht dazu eine einfache Änderung des Wertes, die sofort in den Speicher des Computers übertragen wird. Ein Druck auf <RETURN> ist nicht notwendig. Alle veränderbaren Bytes können mit den Cursor-Tasten erreicht werden. Enthält ein Sektor mehr als 256 Byte (im MFM-Format existieren auch 512- oder 1024-Byte-Sektoren), so wird der Sektorinhalt nach oben »gerollt«, wenn der Cursor über die unterste Zeile hinaus bewegt wird. Ist das Ende des Sektors erreicht, erscheint der Cursor wieder in der obersten Zeile des Fensters. Ebenso erfolgt ein Zurückscrollen des Textes, wenn der Cursor wieder nach oben bewegt wird.

Es werden nur erlaubte Zeichen akzeptiert, womit eine Fehlbedienung des Editors weitgehend ausgeschlossen ist. Wichtig ist in diesem Zusammenhang vielleicht noch: Die Größe des editierbaren Ausschnitts richtet sich immer nach der Größe des aktuellen Sektors. Haben Sie zuletzt einen Sektor mit 512 Byte von einer MFM-Diskette eingelesen und editiert, und wollen diesen in einen 256-Byte-Sektor einer GCR-formatierten Diskette schreiben, so wird der Befehl ausgeführt, die zweite Hälfte des Sektorinhalts wird jedoch abgeschnitten.

Mit dem Drücken von »<« gelangen Sie wieder in den Kommando-Modus des Burstmon, wo Sie den geänderten Sektor zum Beispiel jetzt auf eine Diskette schreiben können.

Der letzte wichtige Befehl des Burstmon wurde schon erwähnt: das @-Zeichen. Geben Sie dieses Zeichen im Kommandomodus ohne Zusatz ein, so erscheint die aktuelle Meldung des Fehlerkanals auf dem Bildschirm, in der Regel »00, OK,00,00«.

Geben Sie hinter dem »@« einen »normalen« Befehl an das Diskettenlaufwerk, zum Beispiel »@n:testformat,tf«, so wird dieser Befehl ausgeführt und die Rückmeldung angezeigt.

Schließlich ist es noch möglich, den Burstmon mit dem @-Befehl auf eine spezielle Laufwerksnummer (8 oder 9) zu fixieren. Das ist zum Beispiel bei der Arbeit mit zwei Diskettenstationen sinnvoll. Hier geben Sie einfach »@8« oder »@9« ein, und schon arbeiten sämtliche Kommandos mit der neuen Gerätenummer.

Arbeiten Sie mit dem Burstmon und machen eine Fehleingabe, so ertönt ein Klingelzeichen und der Burstmon kehrt in den Kommando-Eingabe-Modus zurück und erwartet erneut eine Eingabe. Bei der Fehleingabe von Parametern wird der entsprechende Wert abermals verlangt.

Die U0-Befehle der 1571

Wie Sie beim Abtippen des Burstmon sicherlich bemerkt haben, enthält das Listing eine Reihe von DATA-Zeilen, die ein Maschinenprogramm beinhalten. Dieses Maschinenprogramm ist für die Bedienung des schnellen seriellen Bus im Burst-Modus notwendig, da eine Bearbeitung in Basic zu langsam ist und deshalb nicht erfolgen kann. Wenn Sie sich also genauer mit den U0-Befehlen auseinandersetzen wollen, ist es notwendig, daß Sie über die entsprechenden Maschinesprache-Routinen verfügen, die das Bus-Handling übernehmen. Im Anschluß an die Erklärung

haben wir deshalb die dokumentierten Programme (Listing 2 bis 4) abgedruckt. Sie lassen sich für jeden Zweck beliebig modifizieren und sind dank ihrer Kompaktheit leicht im Speicher des C 128 unterzubringen.

Der Aufbau der U0-Befehle ist im Prinzip recht einfach. Es sind nur einige bestimmte Regeln zu beachten. Generell besteht ein Burst-Kommando (eine weitere Bezeichnung für einen U0-Befehl) aus mindestens 3 Byte. Die ersten beiden Bytes enthalten dabei die hexadezimalen (dezimalen) Werte $55 (85) und $30 (48). Das entspricht den ASCII-Codes für »U0«. Das dritte Byte kennzeichnet den bestimmten U0-Befehl und die Laufwerksnummer (0 für Einzel- oder 0 oder 1 für Doppellaufwerke) anhand der niederwertigen 4 Bits, dem »Low-Nibble«. Die höherwertigen 4 Bits beinhalten bestimmte Flag-Funktionen. Normalerweise schließen sich an das dritte Byte noch weitere Steuer-Bytes an, wobei jeder komplette Befehl aus einer bestimmten Anzahl dieser zusätzlichen Steuer-Bytes besteht.

Generell wird ein U0-Kommando über den Kommandokanal (15) der 1571 übertragen. Sie müssen nur informiert darüber sein, ob das betreffende Burst-Kommando die Floppy 1571 in den schnellen Bus-Modus versetzt oder nicht. Kommt keine Rückmeldung von der Floppystation (zum Beispiel beim Formatier-Kommando), so können Sie direkt nach der Übergabe des U0-Befehls ins Basic zurückspringen. Erfordert der Befehl jedoch die Burst-Übertragung von Daten vom Diskettenlaufwerk zum Computer oder umgekehrt, so müssen Sie das wissen und den Computer nach der Befehlsübergabe für die Burst-Übertragung bereit machen, indem Sie zum Beispiel die hinten beschriebenen Maschinesprache-Routinen aufrufen.

Das Burst-Read-Kommando

Dieses Kommando (Tabelle 2) erlaubt das Lesen eines oder mehrerer Sektoren von einer Diskette, wobei diese sofort zum Computer übertragen werden können. Der Burst-Modus der 1571 arbeitet dabei so schnell, daß eine komplette Spur mit zum Beispiel 21 Sektoren in weniger als einer Sekunde geladen werden kann. Zum Vergleich: Die Floppy 1541 benötigt dazu ungefähr 12 Sekunden.

Wenn Sie die Tabelle 2 studieren, erkennen Sie die Anzahl der Befehls-Bytes mit den entsprechenden Funktionen. Am unteren Ende der Tabelle ist angegeben, was über den schnellen seriellen Bus übertragen wird. In diesem Fall schickt die Floppystation direkt nach dem Erhalt des Befehls (der über den Kommandokanal geschickt wird) und der Analyse der eingelegten Diskette eine Meldung in Form des Burst-Statusbyte (auch Controller-Burst-Status oder einfach Statusbyte genannt) zurück, in der dem Computer mitgeteilt wird, ob betreffs des angeforderten Sektors alles in Ordnung ist.

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Befehlsbyte, die Bits haben folgende Bedeutung:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 0
	Bit 2: immer 0
	Bit 3: immer 0
	Bit 4: Wahl der Diskettenseite unter MFM (0 oder 1)
	Bit 5: nur Pufferübertragung; bei 1 wird dem Computer nur der Pufferinhalt, in dem die gelesenen Sektoren normalerweise stehen, übertragen
	Bit 6: bei 1 werden eventuelle Lesefehler ignoriert
	Bit 7: bei 1 wird ein Sektor nur in den Puffer gelesen und nicht zum Computer übertragen
03	Gewünschte Spurnummer zum Lesen
04	Gewünschte Sektornummer zum Lesen
05	Anzahl der Sektoren, die gelesen werden sollen
06	Spurnummer, auf die der Kopf danach positioniert werden soll. Diese Angabe ist optional
Busverkehr: Senden des Kommandos über den Befehlskanal an die Floppy. Die Floppy sendet darauf jeweils ein Statusbyte und nachfolgend den Sektorinhalt zum Computer.	

Tabelle 2. Die Struktur des Burst-Read-Kommandos

Wurde das Statusbyte gesendet und war in Ordnung, so überträgt die 1571 direkt im Anschluß daran den kompletten Sektor (sofern Bit 7 von Byte 2 auf 0 steht), wobei der Computer natürlich »wissen« muß, wie groß die Anzahl der Datenbytes dieses Sektors ist (dazu gleich mehr bei der Besprechung des Inquire-Disk-Kommandos). Sie können sich die Empfangsroutinen des Computers am Ende dieses Artikels schon einmal analysieren, um alle Vorgänge besser zu verstehen.

Die Steuer-Bits und Bytes sind in ihren Funktionen überwiegend ausführlich genug in der Tabelle 2 erläutert. Lediglich die Bits 5 bis 7 in Byte 2, das Byte 5 und Byte 6 bedürfen einer genaueren Erklärung:

Ist Bit 5 gesetzt (Wert = 1), so erfolgt kein Diskettenzugriff. Die Floppystation geht dann davon aus, daß ein eventueller Sektor schon zu einem früheren Zeitpunkt gelesen wurde und noch im internen Pufferspeicher steht. Es erfolgt deshalb lediglich die Übertragung des Pufferinhalts zum Computer.

Wenn Bit 6 auf 1 steht, dann werden Lesefehler auf der Diskette ignoriert. Die Floppystation überträgt dann das, was Sie einzulesen imstande war.

Bit 7 hängt in seiner Funktion eng mit Bit 5 zusammen. Ist Bit 7 nämlich gesetzt, so wird ein Sektor zwar von der Diskette in den floppyinternen Pufferspeicher gelesen; es erfolgt jedoch keine Übertragung zum Computer. Dieser erhält also nur das Burst-Statusbyte.

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte; dabei haben die Bits folgende Bedeutung:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 1
	Bit 2: immer 0
	Bit 3: immer 0
	Bit 4: Seitenwahl bei MFM-Format (Seite 0 oder 1)
	Bit 5: 1 bedeutet, Computer schreibt nur Daten in den Puffer ab $0300; kein Diskettenzugriff
	Bit 6: 1 bedeutet, Schreib- und Lesefehler ignorieren
	Bit 7: 1 bedeutet, daß nur der Sektor geschrieben wird. Die Floppy erwartet die Daten schon im Puffer.
03	gewünschte Spurnummer für Schreiben
04	gewünschte Sektornummer für Schreiben
05	Anzahl der Sektoren zum Schreiben
07	Nächste Spur für Kopfpositionierung nach Ende des Schreibens.
Busverkehr: Der Computer muß nach dem Senden des Kommandos den Busverkehr auf Burst-Betrieb umstellen und gegebenenfalls die Daten für den Sektor an die Floppy schicken. Nach dem Schreiben eines jeden Sektors gibt die Floppy ein Statusbyte aus.	

Tabelle 3. Die Struktur des Burst-Write-Kommandos

Byte 5 bestimmt die Anzahl der zu lesenden Sektoren nach der Übergabe des Burst-Read-Befehls. Wichtig ist in diesem Zusammenhang der Sektorabstand (»sector interleave factor«), der durch einen weiteren Burst-Befehl gesetzt werden kann, den wir später besprechen. Wollen Sie also auf Spur 1 drei Sektoren einlesen (Beginn bei Sektor 0), und der Sektor-Interleave-Faktor beträgt 5, so liest die 1571 zuerst den Sektor 1,0. Anschließend folgt Sektor 1,5 und zum Schluß Sektor 1,10. Ist Spur 1 zum Beispiel mit 10 Sektoren formatiert und Sie setzen den Interleave-Faktor

auf 1, so sind Sie in der Lage, die komplette Spur mit einem einzigen Burst-Read-Kommando einzulesen. Die Anzahl der zu lesenden Sektoren muß dann natürlich auf 10 stehen. Bei mehreren Sektoren überträgt die 1571 übrigens vor jedem Sektor einmal ein Burst-Statusbyte.

Byte 6 des Burst-Read-Kommandos gestattet die Angabe einer Spurnummer, auf die der Schreib-/Lesekopf positioniert werden soll, nachdem der bisherige Lesevorgang abgeschlossen wurde. Das erlaubt bei einem späteren Lesezugriff ein schnelleres Hochfahren des Laufwerks und dient in erster Linie der Zeitersparnis beim Suchen bestimmter Spuren. Die Angabe von Byte 6 ist optional, das heißt, wünschen Sie keine anschließende Spurpositionierung, so lassen Sie das Byte bei der Befehlsübergabe komplett weg.

An dieser Stelle sei einmalig ein kleines Beispiel für das Lesen und Übertragen des Sektors 18,0 gegeben, damit Sie die Bedeutung der einzelnen Parameter sehen.

Der komplette Befehl lautet nach Bytes aufgeschlüsselt:

Byte	Wert(hex)	Wert(bin)	Wert(dez)	Bedeutung:
00	$55	%01010101	85	»U«
01	$30	%00110000	48	»0«
02	$02	%00000010	2	Sektor lesen und zum Computer übertragen, eventuelle Fehler melden, Diskettenseite 0
03	$12	%00010010	18	Spurnummer
04	$00	%00000000	0	Sektornummer
05	$01	%00000001	1	nur ein einziger Sektor
06	weglassen, es erfolgt keine Positionierung			

Übertragen wird (schematisch, nicht lauffähiges Beispiel):

```
OPEN 1,8,15,"U0"+CHR$(2)+CHR$(18)+CHR$(0)+CHR$(1)
...Sprung zu den Burst-Routinen in Maschinensprache...
CLOSE 1
```

Das Burst-Write-Kommando

Ein Sektor, der sich entweder im Puffer der 1571 oder im Computer befindet, wird auf die Diskette geschrieben. Seine Syntax ist der des Burst-Read-Kommandos sehr ähnlich (Tabelle 3).

Da das Burst-Write-Kommando analog zum Burst-Read arbeitet, besitzen auch die Steuerbits überwiegend die gleiche Bedeutung. Die Unterschiede sind:

In Byte 2 bestimmt Bit 5, ob die Floppystation lediglich die Sektordaten vom Computer erhält und in den internen Puffer ab Adresse $0300 speichert, oder ob sie den Sektor auch tatsächlich auf die Diskette schreibt. Bit 7 gibt im gesetzten Zustand die Anweisung, daß lediglich der schon vorhandene Pufferinhalt auf die Diskette geschrieben wird. Eine Übertragung der Sektordaten vom Computer zur Floppystation erfolgt nicht.

Wie Sie sehen, versetzt Bit 5 Sie in die Lage, auf einfache und schnelle Weise Maschinenprogramme im Puffer der 1571 unterzubringen. Dieser »Mißbrauch« des Burst-Write-Kommandos ist nur ein Beispiel, wie vielfältig die U0-Befehle eingesetzt werden können.

Das Inquire-Disk-Kommando

Dieses schon öfter bekannte Kommando muß nach jedem Einlegen einer neuen Diskette in das Laufwerk der 1571 ausgeführt werden, wenn die U0-Befehle Anwendung finden sollen. Der Grund ist folgender: Wollen Sie bestimmte Schreib-/Lesezugriffe auf eine Diskette durchführen, muß der Diskcontroller vorher darüber informiert werden, wie die Diskette »aussieht«, die eingelegt wurde. Das heißt er erfährt die Anzahl der Sektoren auf jeder Spur, die jeweils kleinste und größte Nummer der Sektoren einer Spur und die Größe der Sektoren (128, 256, 512 oder 1024 Datenbyte pro Sektor). Das ist notwendig, da es sehr viele verschiedene Diskettenformate unter MFM gibt, denn stellen Sie sich einmal vor, der Diskcontroller würde auf Sektoren zugreifen, die es gar nicht gibt und dadurch andere vorhandene Daten löschen...

Aus diesem Grund wird der Floppystation ihr neuer »Arbeitspartner« vorgestellt. Dabei ist sorgfältig zwischen dem Commodore-eigenen GCR-Format und dem internationalen Standard des MFM-Formats zu unterscheiden.

Während das GCR-Format von Commodore immer eine gleiche Struktur auf der Diskette besitzt und deshalb keine spezielle Anpassung benötigt, haben wir gesehen (und werden noch intensiver mitbekommen), daß es sehr viele verschiedene MFM-Formate gibt. Führen Sie also ein Inquire-Disk-Kommando aus, so fährt der Schreib-/Lesekopf der Floppystation auf Spur 1 zurück und liest Sektor 0 ein. Erkennt die 1571 eine GCR-formatierte Diskette, so wird lediglich die ID der Diskette festgestellt und gespeichert (ähnlich dem I-Befehl; das Einlesen der BAM von Sektor 18,0 unterbleibt jedoch). Erkennt die Floppystation keine GCR-formatierte Diskette (Lesefehler von GCR-Diskcontroller), so schaltet sie auf MFM-Betrieb um und versucht jetzt das Format zu analysieren. Hat sie Erfolg, so ist die Diskette »fixiert«. Hat sie keinen Erfolg, so gibt sie eine Fehlermeldung aus. In allen Fällen erfolgt eine Rückmeldung über das Burst-Statusbyte, anhand dessen sich das Ergebnis der Analyse feststellen läßt.

Erscheint beim Arbeiten mit der 1571 und den Burst-Kommandos irgendwann einmal die Fehlermeldung »DISK ID MISMATCH OR DISK CHANGED«, so haben Sie vermutlich das Inquire-Disk-Kommando (Tabelle 4) vergessen.

Das Format-Kommando

Mit diesem Kommando ist es möglich, Disketten sowohl im GCR als auch im MFM-Format zu formatieren. Es handelt sich hierbei jedoch um ein »reines« Formatieren, das heißt es wird kein Directory oder eine andere organisatorische Einrichtung auf die Zieldiskette geschrieben. Die Kommandofolge und Belegung der einzelnen Steuerbytes sehen Sie in Tabelle 5a (Formatieren im MFM-Format) und 5b (Formatieren im GCR-Format).

Da das Format-Kommando sehr umfangreich ist, soll zunächst das MFM-Format erläutert werden.

Wie Sie sehen, haben Sie individuelle Einstellmöglichkeiten. Sie können die Anzahl der Spuren, der Sektoren pro Spur, die Größe der Sektoren und vieles mehr angeben.

Sie können zum Beispiel auch eine Sektortabelle angeben, wenn Sie Wert auf vollkommen durcheinandergewürfelte Sektornummern legen. Entfällt die Sektortabelle, so nimmt die 1571 für jede Spur aufsteigende Nummern für die Sektoren (also zum Beispiel 0, 1, 2, ...). Gehen wir zuerst davon aus, daß Sie ein »normales« Format wünschen.

Alle angegebenen Bytes bis zu Byte 3 sind optional, müssen also nicht angegeben werden. Die 1571 setzt dann Standardwerte. Wollen Sie jedoch den Wert in Byte 8 ändern, so müssen Sie auch die Bytes 4 bis 7 angeben.

Byte 3 ist das generelle Steuerbyte für die Formatierung. Die Bits 0 bis 5 bestimmen dabei den jeweils logischen Startsektor einer Spur. Bit 6 entscheidet, ob eine Sektortabelle vom Computer geliefert wird oder nicht. Bit 7 entscheidet über die Art der Formatierung: 0 bedeutet GCR und 1 bedeutet MFM (in unserem Fall ist es also jetzt 1).

Byte 4 gibt den Hardware-Interleave-Faktor an. Das ist der physikalische Zählfaktor für die Reihenfolge der Sektoren. Er ist optional und steht normalerweise auf 1, das heißt auf einer Spur folgt nach Sektor 0 Sektor 1, Sektor 2 etc.

Byte 5 bestimmt mit den Werten 0 bis 3 die Größe der Sektoren. Voreinstellung: 256 Datenbytes (Byte 5=1).

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte, wobei die Bits folgendermaßen zu setzen sind:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 0
	Bit 2: immer 1
	Bit 3: immer 0
	Bit 4: Seitennummer bei MFM-Diskette
	Bit 5: egal
	Bit 6: egal
	Bit 7: egal
Busverkehr: Computer muß nach dem Senden des Kommandos auf den Burst-Modus umschalten. Die Floppy gibt nach der Ausführung des Befehls ein Statusbyte aus.	

Tabelle 4. Die Struktur des Inquire-Disk-Kommandos. Mit ihm erkennt die Floppy 1571 das Format einer neuen Diskette

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte, wobei die Bits folgende Bedeutung haben:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 1
	Bit 2: immer 1
	Bit 3: immer 0
	Bit 4: Seitennummer der Diskette 0 oder 1
	Bit 5: Flag für Formatieren; 1=beidseitig; 0=einseitig
	Bit 6: Flag für Index Address Mark; 1= schreiben; 0= nicht schreiben
	Bit 7: immer 0 bei MFM
03	Steuerbyte für Formatierung: Bits 0 bis 5 enthalten den logischen Startsektor
	Bit 6: Anzeige der Sektortabelle: 1=Sektortabelle wird vom Computer übergeben 0= Sektortabelle wird von der Floppy selbst erstellt.
	Bit 7: Flag für Formatierung; bei MFM-Format immer 1
04	Sektorabstand für Hardware (sector interleave) (Angabe nicht notwendig); bei fehlender Angabe 0
05	Größe der Sektoren (0-3): (Angabe nicht notwendig) 0—128 Bytes pro Sektor 1—256 Bytes pro Sektor (bei fehlender Angabe) 2—512 Bytes pro Sektor 3—1024 Bytes pro Sektor
06	Letzte Spurnummer beim Formatieren; Angabe nicht notwendig; sie wird dann 39 gesetzt
07	Anzahl der Sektoren pro Spur (Angabe nicht notwendig). Wird der Parameter angegeben, so richtet sich die Maximalzahl der Sektoren nach Byte 05 und zwar: 26 bei Größe 0; 16 bei Größe 1; 9 bei Größe 2; 5 bei Größe 3
08	Nummer der Spur für Start der Formatierung (Angabe nicht notwendig; Floppy setzt dann 0)
09	Physikalischer Beginn der ersten Spur: (Angabe nicht notwendig; Floppy setzt dann 0)
10	Füllbyte; Wert, mit dem die Sektoren beim Formatieren gefüllt werden sollen; (Angabe nicht notwendig; die Floppy setzt dann $e5 als Defaultwert)
11-...	Bytes der Sektortabelle, falls in Byte 03 des Befehls das Bit 6 gesetzt war.
Busverkehr: Nach dem Senden des Befehls und der Daten vom Computer über den Kommandokanal kein Busverkehr mehr. Die Floppy sendet beim Formatieren kein Statusbyte!	

Tabelle 5a. Die Struktur des Format-Kommandos für das MFM-Format

Byte 6 bestimmt die letzte zu formatierende Spur. Bei fehlender Angabe erfolgt eine Formatierung bis Spur 39, was dann den 40 Spuren (0 bis 39) unter MFM entspricht.

Byte 7 gibt die maximale Anzahl von Sektoren pro Spur an. Diese Angabe ist ebenfalls optional und hängt von Byte 5 ab. Je größer die Sektoren, desto weniger passen auf eine Spur (siehe Tabelle 5a)

Byte 8 bestimmt die Startspur für die Formatierung. Normalerweise wird für diesen Wert die 0 eingesetzt.

Byte 9 gibt den physikalischen Beginn der ersten Spur an. Sie können also eine Formatierung verschieben, indem Sie auf der physikalischen Spur 2 beispielsweise mit Spur 0 (Byte 8) beginnen.

Byte 10 bestimmt den Leerinhalt der Sektoren. Beim Formatieren werden die Sektoren mit diesem entsprechenden Byte vollgeschrieben, wobei der Standardwert $E5 beträgt.

Ab Byte 11 steht gegebenenfalls eine komplette Sektortabelle, falls das in Byte 3, Bit 6 gewünscht wurde. Diese Tabelle enthält nur die Sektornummern (zum Beispiel: 0, 1, 2, 3, 4, 4, 3, 2, 1, ...) und wird von der Floppystation auf Korrektheit und Vollständigkeit überprüft. Natürlich müssen Sie dabei wissen, wie viele Sektoren pro Spur angegeben wurden (Byte 7).

Das Formatieren im GCR-Format ist sehr viel einfacher, wie Tabelle 5b beweist. Auf eine zusätzliche Erklärung der Parameter können wir hier getrost verzichten.

Im Gegensatz zu anderen U0-Befehlen erfolgt nach der Ausführung keine Ausgabe eines Burst-Statusbytes. Wartet der Computer also darauf, so wird er »Pech haben« und sich vermutlich »aufhängen«. Sie können das Format-Kommando also auch bequem von Basic aus bedienen. Eine neu formatierte Diskette gilt übrigens noch nicht als »bekannt«. Sie müssen sie also vor dem ersten Zugriff nach dem Formatieren mit dem Inquire-Disk-Befehl »einloggen«.

Das Sector-Interleave-Kommando

Vom Interleave-Faktor haben wir bisher schon gehört. Nun sei einmal genauer erklärt, was damit eigentlich gemeint ist. Immerhin beherrscht die Floppy 1571 theoretisch drei verschiedene!

Beim Format-Kommando lernten Sie den Hardware-Interleave-Faktor kennen. Er bezeichnet den, einmal beim Formatieren festgelegten und dann nicht mehr veränderlichen, logischen Abstand der Sektoren. Steht dieser Wert auf 1, so werden die Sektoren der Reihe nach durchnumeriert, also 0, 1, 2, 3, 4 (bei fünf Sektoren pro Spur). Steht der Wert aber beispielsweise auf 2 und wir haben eine Spur mit fünf Sektoren vor uns, dann lautet die Reihenfolge der Sektoren: 0, 2, 4, 1, 3.

Der Software-Interleave-Faktor, den wir beim Burst-Read-Kommando schon besprochen haben, gibt an, wie viele Sektoren jeweils übersprungen werden, bevor der nächste eingelesen wird. Das gilt natürlich nur, wenn mehrere Sektoren hintereinander bearbeitet werden sollen. Für unser Beispiel heißt also ein Software-Interleave-Faktor von 1 bei einem Hardware-Interleave-Faktor von ebenfalls 1: Es werden nacheinander die Sektoren 0, 1, 2, ... eingelesen. Bei einem Software-Interleave Faktor von 2 bekommen wir jedoch 0, 2, 4, 1, 3.

Der dritte Interleave-Faktor, den die 1571 kennt, ist für uns im Burst-Modus kaum von Bedeutung. Hier handelt es sich um den rein organisatorischen Interleave-Faktor beim Speichern von Dateien auf eine Diskette im GCR-Format. Hier hat es sich beispielsweise so eingebürgert, daß die 1541 einen Wert von 10 besitzt, das heißt: Beginnt eine Datei bei 1,0, dann steht der nächste Datenblock dieser Datei in Sektor 1,10 und so weiter.

Das Sector-Interleave-Kommando (Tabelle 6) erlaubt sowohl das Auslesen des augenblicklich eingestellten Wertes als auch das Setzen eines neuen Faktors. Und zwar han-

delt es sich hierbei um den Software-Interleave-Faktor für das MFM-Format. Wichtig ist in diesem Zusammenhang lediglich die Tatsache, daß die 1571 beim Setzen des Faktors ein Burst-Byte mit dem neuen Wert erwartet und beim Auslesen des Faktors ein Burst-Byte auf den seriellen Bus ausgibt. Bit 7 in Byte 2 entscheidet dabei über beide Kommandovarianten.

Das Query-Disk-Format-Kommando

Bei diesem Kommando (Tabelle 7) handelt es sich um eine erweiterte Version von Inquire-Disk, der eigentlich nur bei MFM-formatierten Disketten interessant ist. Er ermöglicht eine umfassende Analyse der eingelegten Diskette und deren Format, wobei hier im Gegensatz zu Inquire-Disk auch eine spezielle Spur für die Untersuchung angegeben werden kann.

Nach Ausführung des Befehls gibt die Floppystation ein Burst-Statusbyte aus. Wurde bei der eingelegten Diskette MFM-Format festgestellt, bekommt der Anwender zusätzlich folgende Informationen:

2. Burst-Byte: Anzahl der Sektoren der gewählten Spur
3. Burst-Byte: Spurnummer, die im ersten Sektor-Header auf der gewählten physikalischen Spur gefunden wurde. Es handelt sich hierbei um die logische Spurnummer, die sich von der physikalischen unterscheiden kann
4. Burst-Byte: kleinste Sektornummer der Spur
5. Burst-Byte: größte Sektornummer der Spur
6. Burst-Byte: Hardware-Interleave-Faktor, wie er beim Formatieren der Spur gewählt wurde

Beim GCR-Format erfolgt außer dem Burst-Statusbyte keine weitere Ausgabe.

Das Inquire-Status-Kommando

Mit diesem Kommando (Tabelle 8) können Sie das Burst-Statusbyte von der Floppystation abrufen oder es sogar nach eigenem Belieben neu setzen. Dazu ein paar Erläuterungen: Bit 7 in Byte 2 entscheidet, ob ein neues Byte gesetzt oder ein altes gelesen wird (siehe auch Sector-Interleave-Kommando). Bit 6 entscheidet über den Zustand des »Diskette-gewechselt-Flags«. Wird Bit 6 gesetzt und der Inquire-Status abgefragt, so gibt das Status-Byte an, ob ein Diskettenwechsel stattgefunden hat. Ist Bit 6 hingegen gesetzt, und es wird ein neuer Status gesetzt, so wird der 1571 gleichzeitig mitgeteilt, daß ein Diskettenwechsel stattgefunden hat, obwohl das unter Umständen gar nicht der Fall war. Das nächste Kommando, das in diesem Fall vom Anwender kommt, müßte also ein Inquire-Disk oder Query-Disk-Format-Kommando sein.

Das Fastload-Kommando

Dieses Kommando (Tabelle 9) wird vom C 128 dazu verwendet, Programme oder Daten im schnellen Burst-Modus von einer Diskette zu laden. Die Geschwindigkeit ist dabei etwa um den Faktor 9 bis 11 höher als die des C 64 mit einer angeschlossenen 1541.

Bit 7 in Byte 2 bedarf vielleicht einer besonderen Erwähnung. Hier können Sie nämlich entscheiden, ob der Befehl nur nach einer PRG-Datei mit dem angegebenen Namen sucht und diese lädt, oder ob auch ein eventuell vorhandenes SEQ-File schnell geladen werden darf. Ist das Bit gelöscht, und es existiert nur eine SEQ-Datei mit dem angegebenen Dateinamen, so ist ein »FILE TYPE MISMATCH ERROR« die Folge. Das Fastload-Kommando kann nur bei GCR-formatierten Disketten im C 128-Modus verwendet werden.

Burst-Statusbyte kontra Fehlerkanal

Das Burst-Statusbyte

Wie Sie sehen, sind die 8 Bit des Burst-Statusbytes (Tabelle 10) mit den unterschiedlichsten Belegungen verse-

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte, wobei die Bits folgende Bedeutung haben:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 1
	Bit 2: immer 1
	Bit 3: immer 0
	Bit 4: egal
	Bit 5: egal
	Bit 6: egal
	Bit 7: Flag für Formatierung: Hier kann gewählt werden ob eine Spur vor dem Formatieren ausgemessen wird oder nicht. Das Ausmessen sorgt für eine gleichmäßige Verteilung aller Sektoren und entsprechend gleich großen Lücken zwischen den einzelnen Sektoren. 1 bedeutet Spur ausmessen 0 bedeutet Spur nicht ausmessen
03	Steuerbyte für Formatierung: bei GCR immer 0
04	erstes Zeichen der ID
05	zweites Zeichen der ID
	Busverkehr: Auch hier gilt wieder, daß die Floppy kein Burst-Statusbyte ausgibt, nachdem die Diskette formatiert wurde.

Tabelle 5b. Das Format-Kommando für das GCR-Format weist eine einfach zu handhabende Struktur auf

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte; wobei die Bits folgende Bedeutung haben:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 0
	Bit 2: immer 0
	Bit 3: immer 1
	Bit 4: immer 0
	Bit 5: egal
	Bit 6: egal
	Bit 7: Flag für Schreiben oder Lesen des Interleave: 0= Interleave wird neu gesetzt 1= Interleave wird von der Floppy ausgegeben
03	Interleave-Wert (nur, wenn 02/Bit 7 gleich 0)
	Busverkehr: Wird Interleave neu gesetzt, so muß der Computer das Byte hinter den Kommandobytes an die Floppy senden. Die Floppy gibt daraufhin kein Byte aus. Wird der Interleave-Wert vom Computer gelesen, so sendet die Floppy nach Erhalt des Kommandos ein Byte mit dem Interleave-Wert.

Tabelle 6. Die Struktur des Sektor-Interleave-Kommandos. Es ist vor allem beim MFM-Format von Bedeutung.

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte; die Bits haben folgende Bedeutung:
	Bit 0: Drivenummer (immer 0)
	Bit 1: immer 1
	Bit 2: immer 0
	Bit 3: immer 1
	Bit 4: Diskettenseite im MFM-Format
	Bit 5: egal
	Bit 6: egal
	Bit 7: Flag für folgende Spurnummer: bei 1 folgt noch Byte 03 mit Spurnummer
03	Spurnummer (angeben, wenn 02/Bit 7 gleich 1 ist)
	Busverkehr: Nach Beenden des Kommandos gibt die Floppy ein Burst Statusbyte aus. Wurde ein MFM-Format festgestellt, so folgen zusätzlich zum Statusbyte folgende Informationen:

Tabelle 7. Die Struktur des Query-Disk-Format-Kommandos

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte; wobei die Bits folgende Bedeutung haben: Bit 0: Drivenummer (immer 0) Bit 1: immer 0 Bit 2: immer 1 Bit 3: immer 1 Bit 4: immer 0 Bit 5: egal Bit 7,6: 00 — Burst-Statusbyte schreiben (Byte 03) 01 — Burst-Status schreiben und Diskettenwechsel-Flag löschen 10 — Burst-Statusbyte lesen 11 — Burst-Status lesen und Diskettenwechsel darin anzeigen
03	Neues Statusbyte, wenn 02/Bit 7 gleich 0 war
Busverkehr: Die Floppy erwartet vom Computer entweder ein Statusbyte, oder Sie gibt den aktuellen Status nach Erhalt des Kommandos aus.	

Tabelle 8. Die Struktur des Inquire-Status-Kommandos

Byte	Bedeutung
00	entspricht dem Zeichen U mit dem Wert $55
01	entspricht dem Zeichen 0 mit dem Wert $30
02	Kommandobyte; wobei die Bits folgende Bedeutung haben: Bit 0: immer 1 Bit 1: immer 1 Bit 2: immer 1 Bit 3: immer 1 Bit 4: immer 1 Bit 5: egal Bit 6: egal Bit 7: Flag für Programmdatei; ist dieses Bit gesetzt, so muß die angegebene Datei keine Programmdatei sein; es kann auch eine sequentielle Datei sein. Ist das Bit=0, so wird nur nach einer Programmdatei mit dem gegebenen Namen gesucht!
03-...	Name der gesuchten Datei
Busverkehr: Nach Erhalt des Befehls gibt die Floppy jeden Sektor der Datei mit dem Burst-Statusbyte davor aus.	

Tabelle 9. Die Struktur des Fastload-Kommandos

hen, je nachdem was für ein Befehl vorliegt oder welches Format die eingelegte Diskette besitzt.

Der Unterschied zwischen dem Burst-Statusbyte und der »normalen« Fehlermeldung der 1571 über den Kommandokanal ist folgender: Das Burst-Statusbyte wird nur nach einem Burst-Kommando gesetzt und gibt Aufschluß über den Erfolg einer Aktion. Dabei ergibt sich dieses Burst-Statusbyte direkt aus der Rückmeldung des Diskcontrollers an das DOS und zeigt aus diesem Grund den floppyinternen Zustand an.

Die Meldung, die über den Kommandokanal der 1571 geschickt wird, kommt über den »langsamen seriellen Bus« und zeigt dem Anwender, ob ein Kommando (außer den Burst-Kommandos) erfolgreich abgeschlossen wurde. Hier werden, im Gegensatz zum Burst-Statusbyte, auch Fehler (zum Beispiel Syntaxfehler) des Programmierers »bemerkt« und angezeigt. Der Unterschied der beiden »Fehlerdetektoren« kann so definiert werden:

Der Fehlerkanal zeigt an, ob ein Kommando ordnungsgemäß ausgeführt werden konnte, oder ob ein Fehler aufgetreten ist. Das Burst-Statusbyte zeigt den Zustand des Diskcontrollers an, wie er nach dem Ausführen eines Burst-Kommandos aussieht. Dabei kommt die Meldung codiert, in Form eines Bytes, über den schnellen seriellen Bus, während die normale Fehlermeldung im Klartext auf den normalen seriellen Bus ausgegeben wird. Es kann natürlich passieren, daß beide Statusmeldungen aktiviert werden. Das ergibt sich zum Beispiel beim Vergessen des Inquire-Disk-Kommandos. Hier meldet das Burst-Statusbyte einen Fehler vom Diskcontroller: »DISK ID MISMATCH OR DISK CHANGE«, und die Klartext-Fehlermeldung der 1571 registrierte, daß ein Befehl nicht ausgeführt werden konnte, weil »29, DISK ID MISMATCH« festgestellt wurde.

Bit	Bedeutung
0-3	Controller-Status der Floppystation nach einem Diskettenzugriff. Der Status hat dabei drei verschiedene Meldungsarten, je nach Befehl und Diskettenformat. Statusbelegung bei einer Diskette im GCR-Format: Bits 3,2,1,0: 0000: ok 0001: ok 0010: sector not found 0011: no sync 0100: data block not found 0101: data block checksum error 0110: format error 0111: verify error 1000: write protect error 1001: header block checksum error 1010: data extends into next block 1011: disk id mismatch/disk change 1100: n.v. 1101: n.v. 1110: syntax error 1111: no drive present Statusbelegung bei einer Diskette im MFM-Format: Bits 3,2,1,0: 0000: ok 0001: ok 0010: sector not found 0011: no address mark 0100: n.v. 0101: data crc error 0110: format error 0111: verify error 1000: write protect error 1001: header block checksum error 1010: n.v. 1011: disk change 1100: n.v. 1101: n.v. 1110: syntax error 1111: no drive present Statusbelegung bei FASTLOAD-Kommando (gesamtes Byte): Bits 7,6,5,4,3,2,1,0: 00000000 ok 00000001 ok 00000010 file not found 00011111 EOI
4-5	Sektorgröße: Bits 5,4: 00: 128 Bytes Sektoren 01: 256 Bytes Sektoren 10: 512 Bytes Sektoren 11: 1024 Bytes Sektoren
6	Drivenummer; bei der 1570/71 immer 0
7	Format-Modus: 0 = GCR; 1 = MFM Dieses Bit bestimmt die Funktionen der restlichen Bits des Statusbytes, wobei im GCR-Modus die Sektorgröße immer auf 256-Byte-Sektoren stehen muß!

Tabelle 10. Die Belegung des Burst-Statusbyte gibt es eine Vielzahl von Informationen

Die Burstroutinen für den C128

Kommen wir nun, am Schluß unseres Kurses, zu den Burst-Routinen für den C128 (Listing 2, 3 und 4).

Zuerst muß der Befehl an die Floppystation übertragen werden. Da die 1571 auf den Befehl sehr schnell reagiert, muß schon die Übertragung in Maschinensprache geschehen, um dann schnell genug in den Burst-Modus schalten zu können. Gehen wir einmal davon aus, daß der Kommandokanal mit »OPEN 1,8,15« geöffnet wurde. Das Y-Register des Prozessors muß dann die Länge des Befehlsstrings enthalten. Anschließend kann direkt der Aufruf der Routine in Listing 2 erfolgen.

Jetzt wurde der Befehlsstring also übertragen. Nachdem in Maschinensprache die Ausgabe zur Floppystation zurückgesetzt wurde, beginnt diese sofort mit der Ausführung. Die Routine in Listing 3 dient dem Einlesen entweder nur eines Statusbytes oder aber eines gesamten Sektors, wobei die Größe des Sektors in 256-Byte-Blöcken in der Speicherstelle $B0 und die Größe eines Blocks ($80 oder $00 für 128 oder 265 Byte) in $B1 abgelegt sein muß. Diese Angaben sind notwendig, da ein Sektor, wie wir wissen, unterschiedlich groß sein kann. Die Adresse, ab der ein Sektor im Speicher des C128 abgelegt werden soll, steht in $FB und $FC (Low und High-Byte). Wichtig ist außerdem, daß der I/O-Bereich des C128 aktiviert ist (Konfigurationsregister $FF00 mit $00 beschreiben). Nach dem Start der Routine in Bild 2 müssen Sie nur noch auf das Ende warten (und da warten Sie nicht lange), und schon ist der Sektor im Speicher des Computers untergebracht.

Das Holen eines einzelnen Statusbytes, falls kein nachfolgender Sektor kommt, ist in der Routine schon enthalten. Ist zum Beispiel ein Fehler aufgetreten, muß einfach der Rest der Routine übersprungen werden (Label »error«).

Analog zum Lesen eines Sektors erfolgt das Schreiben (Listing 4). Es gibt lediglich eine zusätzliche Speicherstelle in der Zeropage zu beachten: $FD. Dieses Byte wird als Zwischenspeicher für das »Clock-Flag« benötigt und ist für den Anwender ansonsten uninteressant.

Zum Schluß noch ein paar technische Daten: Die Übertragungsroutinen arbeiten mit einer Rate von 20000 Byte pro Sekunde; das sind 120000 Bit/s bei einer Taktfrequenz von 1 MHz im C128. Schalten Sie diesen auf FAST (also 2-MHZ-Taktfrequenz), so können Sie eine komplette Spur einer Diskette in einer einzigen Diskettenumdrehung einlesen. Das entspricht der Zeit von 1/5 Sekunde.

(Karsten Schramm/sk)

```
100 REM BURSTMON 3.1 (C) 1986 BY KARSTEN SCHRAMM      <IIA>
101 FAST : POKE 2603,64: PRINT CHR$(27)"U": BANK
    15: POKE 251,0: POKE 252,11: GOSUB 371             <BFN>
102 CLR : PRINT "{CLR,2HOME,CYAN}";CHR$(14);: POK
    E 54784,26: POKE 54785,2                           <ORS>
103 T$="00": S$="00": POKE DEC("1700"),0               <77L>
104 F$="GCR": W$="0256": K$="21"                       <2NT>
105 CS$="OK": DD$=DS$: DD=DS                           <TDM>
106 BS$="": U=8: D$="0": N$="0": GOSUB 107: GOTO
    109                                                <D20>
107 RESTORE 109: FOR X=1 TO 8: READ X$: KEY X,X$+
    CHR$(13): NEXT : RETURN                            <75I>
108 FOR X=1 TO 8: KEY X,"": NEXT : RETURN              <BLQ>
109 DATA "READ","WRITE","$","@","INQUIRE DISK","Q
    UERY DISK","USER 0","EDIT"                         <VDI>
110 CS=0: GOSUB 308: TRAP 365: GOTO 125: TRAP 365      <IJS>
111 REM UNTERROUTINEN                                  <IUB>
112 WINDOW 2,2,77,3,1: IF CS$="OK" THEN PRINT CHR
    $(14);"{GREY3,HOME} TRACK:"T$"{2SPACE}SECTOR:
    "S$"{2SPACE}FORMAT: "F$"{2SPACE}SECTOR WIDTH:
     "W$" BYTES{2SPACE}KAPACITY: "K$: ELSE PRINT
    "{LIG.RED}";                                       <4Q6>
113 PRINT " CONTROLLER STATUS: ";CS$;"{2SPACE}ON
    SIDE ";D$;"{2SPACE}UNIT#";U"{CYAN}": RETURN        <BRN>
114 WINDOW 2,2,77,3,1: PRINT CHR$(14);"{GREY3,HOM
    E} TRACK:"T$"{2SPACE}SECTOR:"S$"{2SPACE}FORMA
    T: "F$"{2SPACE}SECTOR WIDTH: "W$" BYTES{2SPAC
    E}KAPACITY: "K$                                    <TA4>
115 PRINT "{GREY3} CONTROLLER STATUS: ";CS$;"{2SP
    ACE}ON SIDE ";D$;"{2SPACE}UNIT#";U"{CYAN}": R
    ETURN                                              <F1Q>
116 WINDOW 2,5,77,20,0: RETURN                         <VUR>
117 WINDOW 2,5,77,20,1: RETURN                         <FOR>
118 REM MENUE EINBLENDEN                               <JQF>
119 GOSUB 116: PRINT CHR$(14);"{CLR,BLACK}*** BUR
    STMON V1.0{2SPACE}TABLE OF COMMANDS ***"           <RGK>
120 WINDOW 2,23,77,23,1: PRINT CHR$(14);: INPUT "
    {YELLOW}COMMAND: ";C$: PRINT "{CYAN}": RETURN
                                                       <NTO>
121 WINDOW 2,23,77,23,1: PRINT CHR$(14);: INPUT "
    {YELLOW}TRACK{2SPACE}1{3LEFT}";T$: PRINT "{CY
    AN}": T$=LEFT$("00",2-LEN(T$))+T$: RETURN          <OQO>
122 WINDOW 2,23,77,23,1: PRINT CHR$(14);: INPUT "
    {YELLOW}SECTOR{2SPACE}0{3LEFT}";S$: PRINT "{C
    YAN}": S$=LEFT$("00",2-LEN(S$))+S$: RETURN         <BPM>
123 WINDOW 2,23,77,23,1: PRINT CHR$(14);: INPUT "
    {YELLOW}SIDE{2SPACE}0{3LEFT}";D$: PRINT "{CYA
    N}": D$=LEFT$("0",1-LEN(D$))+D$: RETURN            <QKD>
124 WINDOW 2,23,77,23,1: PRINT CHR$(14);: RETURN       <V9R>
125 REM HAUPTPROGRAMM                                  <IMA>
126 PRINT CHR$(14);"{12SPACE,WHITE}****{2SPACE}BU
    RSTMON V3.1{2SPACE}(C) 1986 BY KARSTEN SCHRAM
    M{2SPACE}****{CYAN}"                               <SQB>
127 PRINT CHR$(27)"M";CHR$(142);                       <TA3>
128 PRINT " U**************************************
    **************************************I"           <6LF>
129 FOR A=1 TO 2: PRINT " -{76SPACE}-": NEXT A         <HSV>
130 PRINT " Q**************************************
    **************************************W"           <RGQ>
131 FOR A=1 TO 16: PRINT " -{76SPACE}-": NEXT A        <OU4>
132 PRINT " J**************************************
    **************************************K"           <UHE>
133 PRINT " U**************************************
    **************************************I"           <ESF>
134 PRINT " -{76SPACE}-"                               <ULH>
135 PRINT " J**************************************
    **************************************K"           <6ME>
136 GOSUB 114                                          <BFP>
137 REM BEFEHLSEINGABE                                 <IMA>
138 C$="": GOSUB 120                                   <BRE>
139 X=1: RESTORE 234                                   <FRE>
140 READ A$: IF LEFT$(C$,1)=LEFT$(A$,1) THEN 143      <EGG>
141 X=X+1: IF A$<>"*" THEN 140                         <20J>
142 PRINT CHR$(7): GOTO 137                            <C57>
143 ON X GOSUB 147,155,163,171,189,202,205,222,22
    8                                                  <JIR>
144 :                                                  <1C5>
145 :                                                  <144>
146 GOTO 137                                           <RU6>
147 REM @                                              <JED>
148 GOSUB 124                                          <BNQ>
149 IF C$="@" THEN BEGIN : IF DD>1 THEN PRINT "{L
    IG.RED}";: ELSE PRINT "{YELLOW}";                  <F68>
150 PRINT CHR$(14);"DISK STATUS: ";DD$;"{CYAN}":
    GET KEY A$: RETURN : BEND                          <MQR>
151 IF C$="@8" THEN U=8: POKE 284,U: PRINT CHR$(1
    4);"{YELLOW}UNIT#";U;"{CYAN}": SLEEP 1: RETUR
    N                                                  <9MS>
152 IF C$="@9" THEN U=9: POKE 284,U: PRINT CHR$(1
    4);"{YELLOW}UNIT#";U;"{CYAN}": SLEEP 1: RETUR
    N                                                  <5NS>
153 BS$=MID$(C$,2): OPEN 1,U,15: PRINT#1,BS$: CLO
    SE 1: C$="@": GOTO 147                             <KJJ>
154 PRINT#1 CHR$(7): RETURN                            <CGG>
155 REM READ                                           <JED>
156 T$="": GOSUB 121: IF VAL(T$)<0 OR VAL(T$)>70
    THEN 156                                           <AR5>
157 S$="": GOSUB 122: IF VAL(S$)<0 OR VAL(S$)>255
     THEN 157                                          <JFE>
158 D$="": GOSUB 123: IF VAL(D$)<0 OR VAL(D$)>1 T
    HEN 158: D=VAL(D$)                                 <20C>
159 GOSUB 112                                          <BRQ>
160 OPEN 1,U,15: BS$="U0"+CHR$(0 OR(16*D))+CHR$(V
    AL(T$))+CHR$(VAL(S$))+CHR$(1): GOSUB 361: SYS
     DEC("1460"),0,1,6: DD=DS: DD$=DS$: CLOSE 1        <OTT>
161 GOSUB 308: GOSUB 112                               <AEU>
162 C$="@": GOSUB 147: IF CS$="OK" THEN GOTO 171:
     ELSE RETURN                                       <95H>
163 REM WRITE                                          <JED>
164 T$="": GOSUB 121: IF VAL(T$)<0 OR VAL(T$)>70
    THEN 164                                           <QB7>
165 S$="": GOSUB 122: IF VAL(S$)<0 OR VAL(S$)>255
     THEN 165                                          <JRC>
166 D$="": GOSUB 123: IF VAL(D$)<0 OR VAL(D$)>1 T
    HEN 166: D=VAL(D$)                                 <68C>
167 GOSUB 112                                          <ARQ>
168 OPEN 1,U,15: BS$="U0"+CHR$(2 OR(16*D))+CHR$(V
    AL(T$))+CHR$(VAL(S$))+CHR$(1): GOSUB 361: SYS
     DEC("1460"),0,2,6: DD=DS: DD$=DS$: CLOSE 1        <CJS>
169 GOSUB 308: GOSUB 112                               <8EM>
170 C$="@": GOSUB 147: IF CS$="OK" THEN GOTO 171:
     ELSE RETURN                                       <95H>
171 M=14: REM EDIT                                     <R37>
172 GOSUB 108: POKE 252,11: GOSUB 124: PRINT "EDI
    T": GOSUB 117: PRINT CHR$(M);                      <6N7>
173 FOR X=0 TO 15: POKE 251,X*16: SYS DEC("14A8")
    : PRINT : NEXT X                                   <FV6>
174 P=2816: PRINT CHR$(27)"S";"{HOME}";                <HOF>
175 GET KEY A$: IF A$="←" THEN 188                     <82Q>
176 IF A$="{UP}" OR A$="{DOWN}" OR A$="{RIGHT}" O
    R A$="{LEFT}" THEN GOTO 179: ELSE IF A$=" " T
    HEN A$="{RIGHT}": GOTO 179                         <KPT>
```

```
177 IF A$=CHR$(13) OR A$=CHR$(141) OR POS(0)>53 T
    HEN PRINT : GOTO 175                              <AFU>
178 IF PEEK(211)=2 AND A$=CHR$(160) THEN M=XOR(M,
    128): GOTO 172: ELSE : GOTO 182                   <FFO>
179 IF PEEK(235)>=20 AND A$="{DOWN}" AND P<2816+W
     THEN P=P+16: POKE 252,INT((P+240)/256): POKE
     251,(P+240)-PEEK(252)*256: PRINT CHR$(27)"V{
    UP}": SYS DEC("14A8"): PRINT "{UP}": GOTO 175     <7BH>
180 IF PEEK(235)<=5 AND A$="{UP}" AND P>2816 THEN
     P=P-16: PRINT CHR$(27)"W";: POKE 252,INT(P/2
    56): POKE 251,P-PEEK(252)*256: PRINT "{HOME}"
    ;: SYS DEC("14A8"): PRINT "{HOME}";: GOTO 175     <EGK>
181 PRINT A$;: GOTO 175                               <KKO>
182 IF((A$>"/" AND A$<":") OR(A$>"@" AND A$<"G"))
     AND POS(0)>6 AND POS(0)<54 THEN BEGIN : Z=PE
    EK(235)-5: S=POS(0)-7: B=P+Z*16+INT(S/3): BN=
    S/1.5                                             <QCJ>
183 IF(BN-INT(BN))=0 THEN NE=1: NN=16: GOTO 185       <AI3>
184 IF(BN-INT(BN))>.6 THEN NE=16: NN=1: ELSE PRIN
    T " ";: GOTO 175                                  <EA1>
185 WE=VAL(A$): IF A$>"@" THEN WE=ASC(A$)-55          <63T>
186 BY=((PEEK(B) AND(NE*15)) OR(NN*WE)): POKE B,B
    Y: PRINT A$;: BEND : GOTO 175                     <206>
187 GOTO 175                                          <B43>
188 GOSUB 107: PRINT CHR$(27)"U" CHR$(27);CHR$(14
    );: RETURN                                        <6CP>
189 REM QUERY DISK                                    <HM6>
190 T$="": GOSUB 121: IF VAL(T$)<0 OR VAL(T$)>70
    THEN 190                                          <HFB>
191 D$="": GOSUB 123: IF D$<>"0" AND D$<>"1" THEN
     191                                              <9GO>
192 D=VAL(D$): OPEN 1,U,15: BS$="U0"+CHR$(138 OR(
    D*16))+CHR$(VAL(T$)): GOSUB 361                   <77D>
193 SYS DEC("1460"),7,3,4: DD=DS: DD$=DS$: CLOSE
    1: GOSUB 308: GOSUB 117                           <E9K>
194 IF F$="MFM" THEN BEGIN                            <BIJ>
195 PRINT "ANALYZING THE DISK BROUGHT THE FOLLOWI
    NG PARAMETERS:": PRINT : PRINT                    <4MT>
196 PRINT "LOGICAL TRACK:";PEEK(DEC("1704")): PRI
    NT                                                <520>
197 PRINT "NUMBER OF SECTORS:";PEEK(DEC("1705")):
     PRINT                                            <90V>
198 PRINT "MINIMUM SECTOR:";PEEK(DEC("1703")): PR
    INT                                               <H7D>
199 PRINT "MAXIMUM SECTOR:";PEEK(DEC("1702")): PR
    INT                                               <1CD>
200 PRINT "MFM-SECTOR INTERLEAVE:";PEEK(DEC("1701
    ")): PRINT : BEND                                 <SVS>
201 GOSUB 112: RETURN                                 <KNE>
202 REM EXIT                                          <JAD>
203 POKE 54784,26: POKE 54785,0: PRINT "{2HOME,CY
    AN,CLR}";CHR$(27)"L";CHR$(27)"S";CHR$(142)        <03F>
204 END                                               <54M>
205 REM USER                                          <JME>
206 GOSUB 124: PRINT "USER 0"                         <OBF>
207 GOSUB 117: PRINT "THE FOLLOWING BURST-COMMAND
    S ARE AVAILABLE:": PRINT                          <SO4>
208 PRINT "1 - BURST-READ (ONLY 1 SECTOR)"            <214>
209 PRINT "2 - BURST-WRITE (ONLY 1 SECTOR)"           <4S9>
210 PRINT "3 - INQUIRE DISK"                          <KSG>
211 PRINT "4 - FORMAT MFM"                            <AQ6>
212 PRINT "5 - FORMAT GCR (NO DIRECTORY)"             <DDS>
213 PRINT "6 - SECTOR{SHF.SPACE}INTERLEAVE"           <CUN>
214 PRINT "7 - QUERY{SHF.SPACE}DISK{SHF.SPACE}FOR
    MAT"                                              <V88>
215 PRINT "8 - INQUIRE STATUS"                        <BL7>
216 PRINT "9 - UTILITY{SHF.SPACE}MODE"                <375>
217 PRINT "X - EXIT"                                  <A8K>
218 PRINT : PRINT : PRINT "PLEASE SELECT (1-9,X):
     ";                                               <S0H>
219 GET KEY A$: IF A$<"1" OR(A$>"9" AND A$<>"X")
    THEN 219                                          <J9H>
220 GOSUB 117: ON VAL(A$)+1 GOTO 221,155,163,222,
    235,249,255,189,267,280                           <190>
221 GOSUB 117: RETURN                                 <4LF>
222 REM INQUIRE DISK                                  <JQF>
223 D$="0": S$="00": T$="00": CS$="OK": F$="GCR":
     W$="0256": K$="21"                               <8T7>
224 OPEN 1,U,15: BS$="U0"+CHR$(4): GOSUB 361          <92L>
225 SYS DEC("1460"),1,3,3: DD=DS: DD$=DS$: CLOSE
    1                                                 <1I2>
226 GOSUB 308: GOSUB 112                              <AMT>
227 RETURN                                            <SN7>
228 REM $                                             <IIA>
229 GOSUB 124: PRINT "DIRECTORY ON UNIT#";U           <06H>
230 GOSUB 117: PRINT CHR$(27)+"L": SLOW : IF U=9
    THEN DIRECTORY ON U9: GOTO 232                    <ECB>
231 DIRECTORY ON U8                                   <HBB>
232 PRINT : PRINT "{YELLOW}PRESS ANY KEY!{CYAN}":
     DD=DS: DD$=DS$: PRINT CHR$(27)+"M": FAST         <QLM>
233 GET KEY A$: GOSUB 117: C$="@": GOTO 147           <4NE>
234 DATA "@","READ","WRITE","EDIT","QUERY DISK","
    X","USER","INQUIRE DISK","$","*"                  <H5K>
235 REM FORMAT MFM                                    <JED>
236 GOSUB 124: A$="": INPUT "INDEX ADDRESS MARK W
    RITTEN (Y/N){2SPACE}Y{3LEFT}";A$: IF A$="Y" T
    HEN BY=(BY OR 64): ELSE IF A$<>"N" THEN 236       <74R>
237 GOSUB 124: A$="": INPUT "FORMAT DOUBLE SIDED
    (Y/N){2SPACE}Y{3LEFT}";A$: IF A$="Y" THEN BY=
    (BY OR 32): GOTO 239: ELSE IF A$<>"N" THEN 23
    7                                                 <SV2>
238 GOSUB 124: A$="": INPUT "WHICH SIDE DO YOU WI
    SH TO FORMAT (0/1){2SPACE}0{3LEFT}";A$: IF A$
    ="1" THEN BY=(BY OR 16): ELSE IF A$<>"0" THEN
     238                                              <HPB>
239 BS$="U0"+CHR$(BY): A$="": GOSUB 124: INPUT "L
    OGICAL STARTING SECTOR (0-63){2SPACE}0{3LEFT}
    ";A$: A=VAL(A$): IF A<0 OR A>63 THEN 239: ELS
    E BS$=BS$+CHR$(A OR 128)                          <P4V>
240 GOSUB 124: A$="": INPUT "SECTOR INTERLEAVE (0
    -255){2SPACE}0{3LEFT}";A$: A=VAL(A$): IF A<0
    OR A>255 THEN 240: ELSE BS$=BS$+CHR$(A)           <7BV>
241 GOSUB 124: A$="": INPUT "SECTOR SIZE (0,1,2,3
    =128,256,512,1024){2SPACE}1{3LEFT}";A$: A=VAL
    (A$): IF A<0 OR A>3 THEN 241: ELSE BS$=BS$+CH
    R$(A)                                             <5L2>
242 GOSUB 124: A$="": INPUT "LAST TRACK NUMBER (0
    -39){2SPACE}39{4LEFT}";A$: A=VAL(A$): IF A<0
    OR A>39 THEN 242: ELSE BS$=BS$+CHR$(A)            <5H2>
243 GOSUB 124: A$="": INPUT "NUMBER OF SECTORS (D
    EPENDS ON SECTOR SIZE)";A$: A=VAL(A$): IF A<0
     OR A>26 THEN 243: ELSE BS$=BS$+CHR$(A)           <PMO>
244 GOSUB 124: A$="": INPUT "LOGICAL STARTING TRA
    CK{2SPACE}0{3LEFT}";A$: A=VAL(A$): IF A<0 OR
    A>39 THEN 244: ELSE BS$=BS$+CHR$(A)               <FT0>
245 GOSUB 124: A$="": INPUT "STARTING TRACK OFFSE
    T{2SPACE}0{3LEFT}";A$: A=VAL(A$): IF A<0 OR A
    >39 THEN 245: ELSE BS$=BS$+CHR$(A)                <7PN>
246 GOSUB 124: A$="": INPUT "FILL BYTE{2SPACE}229
    {5LEFT}";A$: A=VAL(A$): IF A<0 OR A>255 THEN
    246: ELSE BS$=BS$+CHR$(A)                         <5UE>
247 OPEN 1,8,15,BS$: DD=DS: DD$=DS$: CLOSE 1          <DGJ>
248 C$="@": GOTO 147                                  <9QN>
249 REM FORMAT GCR                                    <J6C>
250 BY=6: A$="": GOSUB 124: INPUT "PARTIAL FORMAT
     (Y/N){2SPACE}Y{3LEFT}";A$: IF A$="Y" THEN BY
    =(BY OR 128): ELSE IF A$<>"N" THEN 250            <SPD>
251 BS$="U0"+CHR$(BY)+CHR$(0)                         <HIA>
252 GOSUB 124: A$="": INPUT "CHARACTER FOR ID 1{2
    SPACE}X{3LEFT}";A$: A$=LEFT$(A$,1): BS$=BS$+A
    $                                                 <P7E>
253 GOSUB 124: A$="": INPUT "CHARACTER FOR ID 2{2
    SPACE}X{3LEFT}";A$: A$=LEFT$(A$,1): BS$=BS$+A
    $                                                 <U7M>
254 GOTO 247                                          <RE7>
255 REM SECTOR INTERLEAVE                             <IUB>
256 BY=8: A$=""                                       <VTG>
257 GOSUB 124: INPUT "WANT TO READ OR WRITE INTER
    LEAVE (R/W){2SPACE}R{3LEFT}";A$                   <U3L>
258 IF A$<>"W" AND A$<>"R" THEN PRINT "{CTRL+G}":
     GOTO 257                                         <M0K>
259 IF A$="R" THEN BEGIN : BY=(BY OR 128): A$=""      <KAB>
260 GOSUB 124: PRINT "SECTOR INTERLEAVE IS NOW: "
    ;                                                 <1RK>
261 BS$="U0"+CHR$(BY): GOSUB 361: OPEN 1,U,15: SY
    S DEC("1460"),1,3,3: DD=DS: DD$=DS$: CLOSE 1:
     PRINT PEEK(DEC("00FA")): GET KEY A$: GOTO 26
    6: BEND                                           <47H>
262 A$=""                                             <ELH>
263 GOSUB 124: INPUT "PLEASE ENTER NEW SECTOR INT
    ERLEAVE (DEC 0-255/HEX $00-$FF){2SPACE}5{3LEF
    T}";A$: IF LEN(A$)>3 OR LEN(A$)<1 THEN 263        <JA4>
264 IF LEFT$(A$,1)="$" THEN A$=CHR$(DEC(MID$(A$,2
    ,2))): ELSE A$=CHR$(VAL(A$))                      <FN4>
265 BS$="U0"+CHR$(BY)+A$: GOSUB 361: OPEN 1,U,15:
     SYS DEC("1460"),0,3,4: DD=DS: DD$=DS$: CLOSE
     1: GOTO 266                                      <QNV>
266 RETURN                                            <KU7>
267 REM INQUIRE STATUS                                <IE9>
268 BY=12 : A$=""                                     <IRJ>
269 GOSUB 124: INPUT "WANT TO READ OR WRITE STATU
    S (R/W){2SPACE}R{3LEFT}";A$                       <KLP>
270 IF A$<>"W" AND A$<>"R" THEN PRINT "{CTRL+G}":
     GOTO 269                                         <NOG>
271 IF A$="R" THEN BEGIN : BY=(BY OR 128): A$=""      <C9B>
272 GOSUB 124: INPUT "WANT TO TEST DISK CHANGE (Y
    /N){2SPACE}Y{3LEFT}";A$: IF A$<>"Y" AND A$<>"
    N" THEN PRINT "{CTRL+G}": GOTO 272                <NTQ>
273 IF A$="Y" THEN BY=(BY OR 64)                      <E65>
274 BS$="U0"+CHR$(BY): GOSUB 361: OPEN 1,U,15: SY
    S DEC("1460"),1,3,3: DD=DS: DD$=DS$: CLOSE 1:
     GOSUB 308: GOSUB 112: GOTO 279: BEND             <3GO>
275 A$=""                                             <QN1>
276 GOSUB 124: INPUT "PLEASE ENTER NEW BURST-STAT
    USBYTE (DEC 0-255/HEX $00-$FF){2SPACE}0{3LEFT
    }";A$: IF LEN(A$)>3 OR LEN(A$)<1 THEN 276         <100>
277 IF LEFT$(A$,1)="$" THEN A$=CHR$(DEC(MID$(A$,2
    ,2))): ELSE A$=CHR$(VAL(A$))                      <DNC>
278 BS$="U0"+CHR$(BY)+A$: GOSUB 361: OPEN 1,U,15:
     SYS DEC("1460"),0,3,4: DD=DS: DD$=DS$: CLOSE
     1: GOTO 279                                      <SJP>
279 RETURN                                            <SJ7>
280 REM UTILITY MODE                                  <H24>
281 GOSUB 124: PRINT "UTILITY MODE"                   <CDB>
282 GOSUB 117: PRINT "PLEASE SELECT ONE OF THE FO
    LLOWING COMMANDS:": PRINT                         <I9E>
283 PRINT "1 - SET SECTOR INTERLEAVE FOR GCR WRIT
    TEN DISK"                                         <Q53>
284 PRINT "2 - SET THE NUMBER OF RETRIES IN CASE
    OF FAILED JOB"                                    <OQ1>
285 PRINT "3 - ANALYSIS OF ROM SIGNATURE"             <29L>
286 PRINT "4 - MODE SELECT (1541 OR 1570/71)"         <BFE>
287 PRINT "5 - HEAD SELECT (1541 MODE ONLY)"          <EF2>
288 PRINT "6 - SET DEVICE NUMBER"                     <KGF>
289 PRINT "X - EXIT OPTION"                           <V7Q>
290 GET KEY A$: IF A$<"1" OR (A$>"6" AND A$<>"X")
     THEN 290                                         <D81>
```

Listing 1. Der »Burstmon«; ein leistungsfähiges Werkzeug für die »Erforschung« der Floppy 1571

```
291 ON VAL(A$)+1 GOTO 307,292,294,296,298,300,302       <0NL>
292 GOSUB 124: INPUT "GCR SECTOR INTERLEAVE{2SPAC
    E}6{3LEFT}";A$: A=VAL(A$): IF A<1 OR A>255 TH
    EN 292                                               <0G7>
293 BS$="U0>S"+CHR$(A): GOTO 305                         <G0J>
294 GOSUB 124: INPUT "NUMBER OF RETRIES{2SPACE}5{
    3LEFT}";A$: A=VAL(A$): IF A<0 OR A>255 THEN 2
    94                                                   <KC2>
295 BS$="U0>R"+CHR$(A): GOTO 305                         <J8I>
296 GOSUB 124: PRINT "PLEASE WAIT A MOMENT ..."          <BBG>
297 BS$="U0>T": GOTO 305                                 <RLV>
298 GOSUB 124: INPUT "1570/71 MODE OR 1541 MODE (
    1/0){2SPACE}1{3LEFT}";A$: IF A$<>"0" AND A$<>
    "1" THEN 298                                         <G6I>
299 BS$="U0>M"+A$: GOTO 305                              <5EQ>
300 GOSUB 124: INPUT "HEAD 0 OR 1 (1/0){2SPACE}0{
    3LEFT}";A$: IF A$<>"0" AND A$<>"1" THEN 300          <GMT>
301 BS$="U0>H"+A$: GOTO 305                              <N1Q>
302 GOSUB 124: INPUT "DEVICE NUMBER (8-30){2SPACE
    }8{3LEFT}";A$: IF VAL(A$)<8 OR VAL(A$)>30 THE
    N 302                                                <AKJ>
303 BS$="U0>"+CHR$(Z)                                    <34J>
304 OPEN 1,U,15,BS$: CLOSE 1: RETURN                     <M21>
305 OPEN 1,U,15,BS$: DD=DS: DD$=DS$: CLOSE 1             <P98>
306 GOSUB 117: C$="@": GOTO 147                          <PLV>
307 RETURN                                               <SN7>
308 REM BURST-MELDUNGEN AUSWERTEN                        <IIA>
309 DATA "OK"                                            <S54>
310 DATA "OK"                                            <T94>
311 DATA "SECTOR NOT FOUND"                              <47B>
312 DATA "NO SYNC"                                       <NQD>
313 DATA "DATA BLOCK NOT FOUND"                          <VM7>
314 DATA "DATA BLOCK CHECKSUM ERROR"                     <S63>
315 DATA "FORMAT ERROR"                                  <ICR>
316 DATA "VERIFY ERROR"                                  <8UT>
317 DATA "WRITE PROTECT ERROR"                           <3GT>
318 DATA "HEADER BLOCK CHECKSUM ERROR"                   <I30>
319 DATA "DATA EXTENDS INTO NEXT BLOCK"                  <ALH>
320 DATA "DISK ID MISMATCH/DISK CHANGE"                  <50S>
321 DATA "NO DEFINED ERROR MESSAGE"                      <E6U>
322 DATA "NO DEFINED ERROR MESSAGE"                      <M6U>
323 DATA "SYNTAX ERROR"                                  <60R>
324 DATA "NO DRIVE PRESENT"                              <2MA>
325 :                                                    <040>
326 DATA "OK"                                            <T94>
327 DATA "OK"                                            <TD4>
328 DATA "SECTOR NOT FOUND"                              <S8B>
329 DATA "NO ADDRESS MARK"                               <S75>
330 DATA "NO DEFINED ERROR MESSAGE"                      <M6U>
331 DATA "DATA CRC ERROR"                                <95C>
332 DATA "FORMAT ERROR"                                  <JST>
333 DATA "VERIFY ERROR"                                  <BEV>
334 DATA "WRITE PROTECT ERROR"                           <28T>
335 DATA "HEADER BLOCK CHECKSUM ERROR"                   <GB9>
336 DATA "NO DEFINED ERROR MESSAGE"                      <62U>
337 DATA "DISK CHANGE"                                   <61B>
338 DATA "NO DEFINED ERROR MESSAGE"                      <MEV>
339 DATA "NO DEFINED ERROR MESSAGE"                      <UEV>
340 DATA "SYNTAX ERROR"                                  <185>
341 DATA "NO DRIVE PRESENT"                              <AVA>
342 IF PEEK(DEC("1700"))=128 THEN BEGIN : GOSUB 1
    17: PRINT "{8DOWN,LIG.RED}CAUTION!!! THE CONN
    ECTED DRIVE IS NOT IN THE 1570/71 MODE!"             <D4J>
343 PRINT : PRINT : PRINT "{CTRL+G,YELLOW}PRESS A
    NY KEY TO CONTINUE{CYAN}": GET KEY A$                <UDT>
344 ABS ICHTLICHER FEHLER IN DIESER ZEILE                <74U>
345 BEND                                                 <OR6>
346 CS=PEEK(DEC("00FA"))                                 <AFV>
347 IF CS>127 THEN F$="MFM": RESTORE 326: ELSE F$
    ="GCR": RESTORE 309                                  <UD3>
348 N$=STR$(SGN(CS AND 64))                              <7OV>
349 IF F$="MFM" THEN BEGIN                               <KHF>
350 IF(CS AND 48)=0 THEN W=0: W$="0128": POKE 176
    ,1: POKE 177,128: K$="26"                            <5AD>
351 IF(CS AND 48)=16 THEN W=0: W$="0256": POKE 17
    6,1: POKE 177,0: K$="16"                             <7J5>
352 IF(CS AND 48)=32 THEN W=256: W$="0512": POKE
    176,2: POKE 177,0: K$="09"                           <TID>
353 IF(CS AND 48)=48 THEN W=768 : W$="1024": POKE
     176,4: POKE 177,0: K$="05"                          <9AD>
354 BEND : ELSE BEGIN : W=0: W$="0256": POKE 176,
    1: POKE 177,0: T=VAL(T$): IF T>35 THEN T=T-35        <IDD>
355 IF T<36 THEN K$="17"                                 <S9A>
356 IF T<31 THEN K$="18"                                 <O96>
357 IF T<25 THEN K$="19"                                 <U96>
358 IF T<18 THEN K$="21": BEND                           <D60>
359 FOR X=0 TO(CS AND 15): READ CS$: NEXT X              <0I0>
360 RETURN                                               <4S7>
361 REM BEFEHLSSTRING IN SPEICHER SCHREIBEN              <I68>
362 FOR X=1 TO LEN(BS$)                                  <CM8>
363 POKE 5167+X,ASC(MID$(BS$,X,1)): NEXT X: POKE
    251,0: POKE 252,11                                   <D0G>
364 RETURN                                               <4O7>
365 REM FEHLER                                           <IMA>
366 IF ER=14 AND EL=20468 THEN PRINT "{CTRL+G}":
    RESUME 276                                           <8JL>
367 IF ER=14 AND EL=20368 THEN PRINT "{CTRL+G}":
    RESUME 263                                           <8NM>
368 IF ER=11 THEN RESUME 137                             <JMB>
369 IF ER=5 THEN RESUME 142                              <G22>
370 RESUME NEXT                                          <56P>
371 REM MASCHINENPROGRAMM INITIALISIEREN                 <IE9>
372 PRINT "{CLR}DATAS WERDEN INITIALISIERT!"             <17S>
373 RESTORE 376                                          <D7V>
374 FOR X=4864 TO 5373: READ A: POKE X,A: NEXT X         <NBL>
375 RETURN                                               <SR7>
376 DATA 120 , 44 , 13 , 220 , 166 , 176 , 173 ,
    0 , 221 , 73                                         <DCR>
377 DATA 16 , 141 , 0 , 221 , 169 , 8 , 44 , 13 ,
    220 , 240                                            <IQ0>
378 DATA 251 , 173 , 0 , 221 , 73 , 16 , 141 , 0
    , 221 , 173                                          <43I>
379 DATA 12 , 220 , 133 , 250 , 41 , 15 , 201 , 2
    , 176 , 34                                           <28S>
380 DATA 160 , 0 , 169 , 8 , 44 , 13 , 220 , 240
    , 251 , 173                                          <7DM>
381 DATA 0 , 221 , 73 , 16 , 141 , 0 , 221 , 173
    , 12 , 220                                           <7KR>
382 DATA 145 , 251 , 200 , 196 , 177 , 208 , 231
    , 230 , 252 , 202                                    <PGU>
383 DATA 208 , 226 , 24 , 36 , 56 , 88 , 96 , 234
    , 234 , 234                                          <HGL>
384 DATA 120 , 169 , 64 , 133 , 253 , 166 , 176 ,
    160 , 0 , 173                                        <1T4>
385 DATA 5 , 213 , 9 , 8 , 141 , 5 , 213 , 169 ,
    127 , 141                                            <HQN>
386 DATA 13 , 220 , 169 , 0 , 141 , 5 , 220 , 169
    , 3 , 141                                            <B59>
387 DATA 4 , 220 , 173 , 14 , 220 , 41 , 128 , 9
    , 85 , 141                                           <M53>
388 DATA 14 , 220 , 44 , 13 , 220 , 173 , 0 , 221
    , 205 , 0                                            <VEO>
389 DATA 221 , 208 , 248 , 69 , 253 , 41 , 64 , 2
    40 , 242 , 165                                       <DNQ>
390 DATA 253 , 73 , 64 , 133 , 253 , 177 , 251 ,
    141 , 12 , 220                                       <SP5>
391 DATA 169 , 8 , 44 , 13 , 220 , 240 , 251 , 20
    0 , 196 , 177                                        <KIQ>
392 DATA 208 , 219 , 230 , 252 , 202 , 208 , 214
    , 169 , 8 , 141                                      <2UQ>
393 DATA 14 , 220 , 173 , 5 , 213 , 41 , 247 , 14
    1 , 5 , 213                                          <F2C>
394 DATA 44 , 13 , 220 , 173 , 0 , 221 , 9 , 16 ,
    141 , 0                                              <C85>
395 DATA 221 , 169 , 8 , 44 , 13 , 220 , 240 , 25
    1 , 173 , 12                                         <9FI>
396 DATA 220 , 133 , 250 , 173 , 0 , 221 , 41 , 2
    39 , 141 , 0                                         <406>
397 DATA 221 , 165 , 250 , 41 , 15 , 201 , 2 , 17
    6 , 2 , 24                                           <57I>
398 DATA 36 , 56 , 88 , 96 , 120 , 170 , 240 , 67
    , 44 , 13                                            <M2H>
399 DATA 220 , 173 , 0 , 221 , 73 , 16 , 141 , 0
    , 221 , 169                                          <917>
400 DATA 8 , 44 , 13 , 220 , 240 , 251 , 173 , 0
    , 221 , 73                                           <3A3>
401 DATA 16 , 141 , 0 , 221 , 173 , 12 , 220 , 13
    3 , 250 , 234                                        <E08>
402 DATA 16 , 33 , 41 , 15 , 201 , 2 , 176 , 29 ,
    202 , 240                                            <863>
403 DATA 24 , 169 , 8 , 44 , 13 , 220 , 240 , 251
    , 173 , 0                                            <QJF>
404 DATA 221 , 73 , 16 , 141 , 0 , 221 , 173 , 12
    , 220 , 157                                          <C5E>
405 DATA 0 , 23 , 76 , 10 , 20 , 24 , 36 , 56 , 8
    8 , 96                                               <5HS>
406 DATA 0 , 0 , 0 , 0 , 32 , 32 , 32 , 32 , 32 ,
    32                                                   <9U7>
407 DATA 32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,
    32 , 32                                              <I3E>
408 DATA 32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,
    32 , 32                                              <HR1>
409 DATA 32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,
    32 , 32                                              <HJ0>
410 DATA 32 , 32 , 32 , 32 , 32 , 32 , 32 , 32 ,
    32 , 32                                              <JBB>
411 DATA 32 , 32 , 72 , 138 , 72 , 173 , 28 , 10
    , 41 , 191                                           <TFE>
412 DATA 141 , 28 , 10 , 162 , 1 , 32 , 201 , 255
    , 162 , 0                                            <BAG>
413 DATA 189 , 48 , 20 , 32 , 210 , 255 , 232 , 1
    36 , 208 , 246                                       <0P5>
414 DATA 32 , 204 , 255 , 104 , 170 , 104 , 160 ,
    0 , 140 , 0                                          <C21>
415 DATA 23 , 44 , 28 , 10 , 80 , 15 , 202 , 208
    , 3 , 76                                             <BEI>
416 DATA 0 , 19 , 202 , 208 , 3 , 76 , 80 , 19 ,
    76 , 224                                             <0OL>
417 DATA 19 , 160 , 128 , 140 , 0 , 23 , 96 , 0 ,
    0 , 0                                                <R5K>
418 DATA 0 , 0 , 0 , 0 , 169 , 46 , 32 , 210 , 25
    5 , 165                                              <HPF>
419 DATA 252 , 56 , 233 , 11 , 170 , 165 , 251 ,
    32 , 159 , 184                                       <4FU>
420 DATA 32 , 125 , 255 , 32 , 0 , 160 , 0 , 177
    , 251 , 32                                           <FDN>
421 DATA 194 , 184 , 169 , 32 , 32 , 210 , 255 ,
    200 , 192 , 16                                       <QBU>
422 DATA 144 , 241 , 32 , 125 , 255 , 32 , 32 , 1
    8 , 0 , 160                                          <BJF>
423 DATA 0 , 177 , 251 , 72 , 41 , 127 , 201 , 32
    , 104 , 176                                          <1RK>
424 DATA 2 , 105 , 64 , 32 , 210 , 255 , 200 , 19
    2 , 16 , 208                                         <IRS>
425 DATA 236 , 169 , 0 , 133 , 244 , 133 , 245 ,
    169 , 146 , 76                                       <8QP>
426 DATA 210 , 255 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,
    0                                                    <5I3>
```

Listing 1. »Burstmon« (Schluß)

```
        lda $0a1c     ; Flags für Diskbetrieb laden
        and #$bf      ; Flag für schnellen Busbetrieb löschen
        sta $0a1c     ; und wieder abspeichern
        ldx #$01      ; Dateinummer (in diesem Fall 1)
        jsr $ffc9     ; Ausgabegerät setzen
        ldx #$00      ; Index-Startwert laden; der Monitor legt
                      ; den
loop    lda $1430,x   ; Befehlsstring ab $1430 ab
        jsr $ffd2     ; Zeichen an die Floppy ausgeben
        inx           ; Zeiger erhöhen
        dey           ; Anzahl der Bytes minus 1
        bne loop      ; weiter, wenn noch Zeichen auszugeben
                      ; sind
        jsr $ffcc     ; Ausgabe zur Floppy zurücksetzen
        bit $0a1c     ; Flag für schnellen Busmodus testen
        bvc error     ; Ende, wenn Floppy nicht im Fast-Modus
        ...           ; ...
```

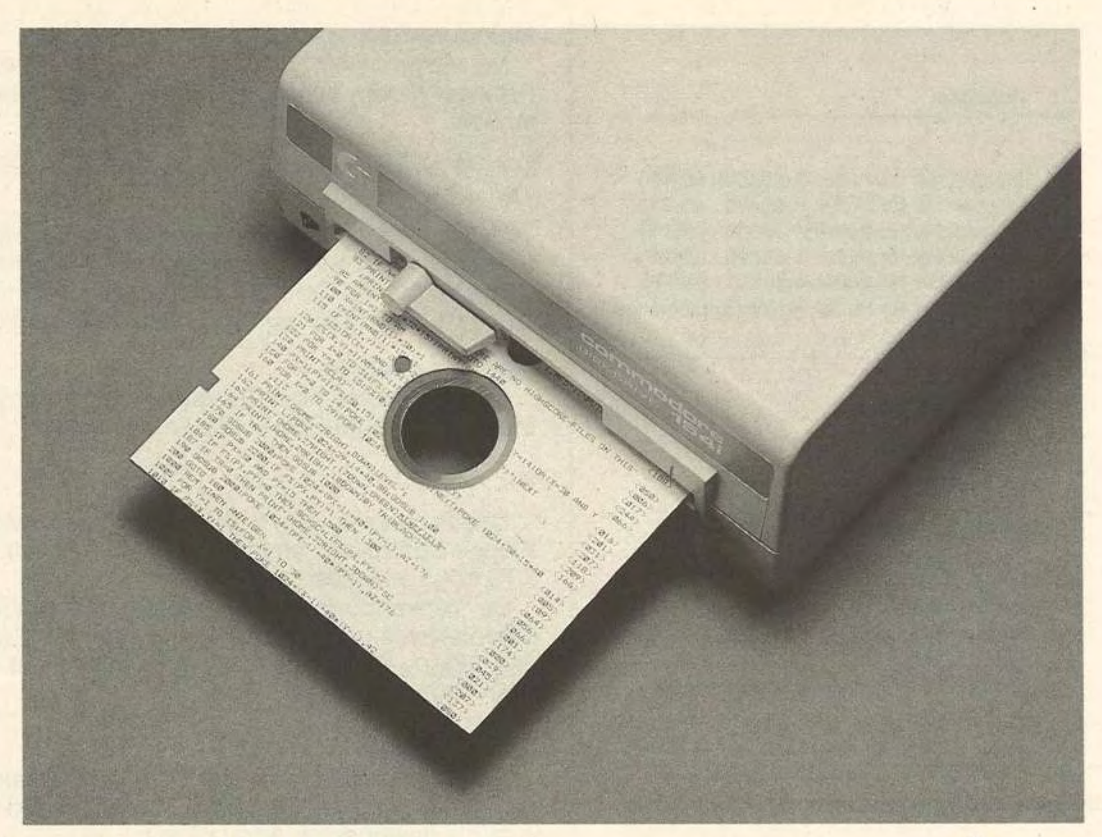
Listing 2. So wird ein Burst-Kommandostring an die 1571 übergeben. Der String steht ab $1430 im Speicher und wird von dort aus an die Floppystation geschickt. Vor dem Aufruf der Routine muß der Kommandokanal der 1571 geöffnet sein.

```
        sei           ; Interrupts unterbinden
        bit $dc0d     ; Kontrollregister löschen
        ldx $b0       ; Anzahl der Blöcke holen
        lda $dd00     ; Register für seriellen Bus lesen
        eor #$10      ; CLOCK-Leitung invertieren
        sta $dd00     ; und setzen
        lda #$08      ; Flag für Schieberegister eingelesen
l1      bit $dc0d     ; testen; Byte erhalten?
        beq l1        ; warten, bis Byte eingelesen
        lda $dd00     ; Register für seriellen Bus lesen
        eor #$10      ; CLOCK-Leitung invertieren
        sta $dd00     ; und wieder setzen
        lda $dc0c     ; Byte aus Schieberegister holen
        sta $fa       ; und als Statusbyte merken
        and #$0f      ; Statusbits isolieren
        cmp #$02      ; Fehlermeldung?
        bcs error     ; verzweige, wenn ja
        ldy #$00      ; Index setzen
loop    lda #$08      ; Flag für Schieberegister eingelesen
l2      bit $dc0d     ; testen; Byte erhalten?
        beq l2        ; warten, bis Byte eingelesen
        lda $dd00     ; Register für seriellen Bus lesen
        eor #$10      ; CLOCK-Leitung invertieren
        sta $dd00     ; und setzen
        lda $dc0c     ; Byte aus Schieberegister holen
        sta ($fb),y   ; und in Puffer schreiben
        iny           ; nächstes Byte
        cpy $b1       ; schon ein Block gelesen?
        bne loop      ; weitermachen, wenn nein
        inc $fc       ; Pufferadresse Hi erhöhen
        dex           ; Anzahl der zu lesenden Blöcke
                      ; vermindern
        bne loop      ; und weiterlesen, bis alles übertragen
        clc           ; Flag für Fehler löschen
        .byte $24     ; nächsten Befehl überspringen
error   sec           ; Flag für Fehler setzen
        cli           ; Interrupts wieder zulassen
        rts           ; Ende
```

Listing 3. Die Routine für das Burst-Read-Kommando. Hier wird ein Sektor von einer Diskette eingelesen und vom Computer empfangen. Im Artikel ist beschrieben, welche Bedeutung den einzelnen Zeropage-Speicherstellen dabei zukommt.

```
        sei           ; Interrupts unterbinden
        lda #$40      ; Flag für CLOCK-Leitung
        sta $fd       ; setzen
        ldx $b0       ; Anzahl der zu schreibenden Blöcke
        ldy #$00      ; Index setzen
        lda $d505     ; MMU Modus-Konfigurationsregister
                      ; lesen
        ora #$08      ; seriellen Bus auf Ausgabe schalten
        sta $d505     ; und neuen Modus setzen
        lda #$7f      ; serielles Schieberegister
        sta $dc0d     ; auf Ausgabe schalten und Flags löschen
        lda #$00      ; Wert für Timer Hi
        sta $dc05     ; setzen
        lda #$03      ; Wert für Timer Lo; Übertragungsrate
                      ; 6 µs
        sta $dc04     ; setzen; schnellstmögliche Über-
                      ; tragungsrate
        lda $dc0e     ; Kontrollregister für Timer lesen
        and #$80      ; Schieberegister auf Ausgang und
        ora #$55      ; Timer auf "free running mode"
        sta $dc0e     ; setzen
        bit $dc0d     ; Flags für Schieberegister löschen
loop    lda $dd00     ; seriellen Bus lesen
        cmp $dd00     ; konstanter Wert?
        bne loop      ; nein, konstanten Wert abwarten
        eor $fd       ; CLOCK-Bit invertieren
        and #$40      ; und prüfen
        beq loop      ; warten, bis Floppy Daten anfordert
        lda $fd       ; CLOCK-Bit
        eor #$40      ; invertieren für nächsten Zustand
        sta $fd       ; und als Flag wieder setzen
        lda ($fb),y   ; Byte aus Puffer holen
        sta $dc0c     ; und ins Schieberegister schreiben
        lda #$08      ; Bit für Schieberegister
l2      bit $dc0d     ; testen; Byte schon ausgegeben?
        beq l2        ; warten, bis Byte ausgegeben
        iny           ; Index auf nächstes Zeichen
        cpy $b1       ; schon ein Block ausgegeben?
        bne loop      ; weitermachen, wenn nein
        inc $fc       ; Pufferadresse Hi erhöhen
        dex           ; Anzahl der auszugebenden Blöcke
                      ; vermindern
        bne loop      ; weitermachen, bis alles ausgegeben
        lda #$08      ; Flag für Timer nach Herunterzählen
                      ; anhalten
        sta $dc0e     ; setzen
        lda $d505     ; Modus-Konfigurationsregister der MMU
        and #$f7      ; seriellen Bus auf Eingang
        sta $d505     ; neuen Modus setzen
        bit $dc0d     ; Flag für Schieberegister löschen
        lda $dd00     ; seriellen Bus lesen
        eor #$10      ; CLOCK-Leitung umschalten
        sta $dd00     ; und wieder setzen
        lda #$08      ; Flag für Schieberegister
l3      bit $dc0d     ; testen; Byte eingelesen?
        beq l3        ; warten, bis Byte eingelesen
        lda $dc0c     ; Byte aus Schieberegister holen
        sta $fa       ; als Statusbyte merken
        lda $dd00     ; seriellen Bus lesen
        and #$ef      ; CLOCK-Leitung löschen
        sta $dd00     ; und setzen
        lda $fa       ; Statusbyte zurückholen
        and #$0f      ; Statusbits isolieren
        cmp #$02      ; Fehler aufgetreten?
        cli           ; Interrupts wieder zulassen
        bcs error     ; verzweige zur Fehlerroutine bei Fehler
        rts           ; sonst Ende; alles ok
error   ...           ; Fehlerbehandlung...
```

Listing 4. Die Routine für das Schreiben eines Sektors zur Floppystation. Bitte beachten Sie auch hier die Hinweise im Text.



Floppy 1541 programmieren

Die Commodore-Floppy VC 1541 ist eine »intelligente« Diskettenstation. Ihr mechanisches und elektronisches Innenleben wird von einem eigenen 6502-Mikroprozessor regiert, der sogar vom C64 aus programmiert werden kann.

Durch diese »Intelligenz« bietet sich ein Einsatz des Laufwerks als »Host-Computer« geradezu an. Das heißt, man könnte, während im C64 das Hauptprogramm abläuft, vom Prozessor der Floppy parallel zeitaufwendige Teilaufgaben ausführen lassen, deren Ergebnisse bei Bedarf übernommen würden. Die denkbaren Aufgaben beschränken sich dabei keineswegs allein auf den floppyspezifischen Bereich. Sinnvolle Einsatzmöglichkeiten wären zum Beispiel im Bereich der Grafik die Berechnung komplizierter Funktionswerte, etwa von skalierten 3D-Translations- und Rotationsmatrizen, während sich der C64 ausschließlich um die Ermittlung der Eingangsdaten und die Ausgabe auf dem Bildschirm kümmert...

Da als Mikrocomputer in der Floppy eine 6502-CPU arbeitet, die ja direkt softwarekompatibel zum 6510, dem Prozessor im C64 ist, braucht man sich nicht erst mit den speziellen Eigenarten eines neuen Bausteins vertraut zu machen, ja es lassen sich sogar die vorhandenen Assembler und Utilities des C64 verwenden. Aus diesem Grund dürfte jeden, der sich bereits etwas mit Maschinenprogrammierung auskennt, natürlich die Möglichkeit reizen, aktiv in die floppy-internen Abläufe einzugreifen. Die Entwickler des Laufwerks haben dies scheinbar vorausgeahnt, denn sie spendierten sogar einen eigenen »Benutzerpuffer« in dem ansonsten mit ganzen zwei Kilobyte nicht gerade üppig ausgefallenen RAM-Bereich. Aber bei entsprechender Umsicht lassen sich zumindest auch die anderen vier Puffer mit ihren jeweils 256 Byte mitbenutzen.

Adressbereich	Verwendung
$0000 - $00FF	Arbeitsspeicher des DOS
$0100 - $0144	Stackbereich
$0145 - $01FF	Arbeitsspeicher des DOS
$0200 - $02FF	Error- und Kommandopuffer des DOS
$0300 - $03FF	Puffer 0: Arbeitspuffer (aktueller Block)
$0400 - $04FF	Puffer 1: aktueller Block des Directorys
$0500 - $05FF	Puffer 2: Benutzerpuffer, normal frei
$0600 - $06FF	Puffer 3: letzter Block des Directorys
$0700 - $07FF	Puffer 4: enthält Block 18,0 (BAM)

Tabelle 1. Die Belegung des Floppy-RAM

Die 1541 als »Host-Computer«

Diese Speichergröße hört sich zwar im Zeitalter der Megabit-Chips zunächst lächerlich klein an; die Erfahrung zeigt aber, daß bereits dieser geringe Platz für die meisten Anwendungen durchaus genügt, zumal sich ja viele Routinen aus dem 16 KByte umfassenden Betriebssystem der Floppy im ROM mitbenutzen lassen.

Der gesamte RAM-Bereich der 1541 umfaßt 2 KByte und belegt die Adressen $0000 bis $0800, also dezimal 0 bis 2048. Tabelle 1 zeigt die Einteilung dieses Bereiches in Funktionsgruppen. Hier wird bereits deutlich die Aufteilung in Puffer sichtbar, die bei der 1541, die wir noch sehen werden, eine wichtige Rolle spielen.

Eigentlich hat die Laufwerkselektronik gleich mehrere ganz verschiedenartige Aufgaben gleichzeitig zu bewältigen, nämlich:
1. Bedienung des seriellen IEC-Busses,
2. Auswertung der Befehle,
3. Senden oder Empfangen von Daten,
4. Ein-/Ausschalten des Drivemotors,
5. Positionieren des Schreib-/Lesekopfes.

Adresse	Funktion
$0000	Jobspeicher für Puffer 0 ($0300 - $03FF)
$0001	Jobspeicher für Puffer 1 ($0400 - $04FF)
$0002	Jobspeicher für Puffer 2 ($0500 - $05FF)
$0003	Jobspeicher für Puffer 3 ($0600 - $06FF)
$0004	Jobspeicher für Puffer 4 ($0700 - $07FF)
$0005	Jobspeicher für Puffer 5 (nicht implementiert)
$0006	Track für Puffer 0
$0007	Sektor für Puffer 0
$0008	Track für Puffer 1
$0009	Sektor für Puffer 1
$000A	Track für Puffer 2
$000B	Sektor für Puffer 2
$000C	Track für Puffer 3
$000D	Sektor für Puffer 3
$000E	Track für Puffer 4
$000F	Sektor für Puffer 4
$0010	Track für Puffer 5 (nicht implementiert)
$0011	Sektor für Puffer 5 (nicht implementiert)

Tabelle 2. Bedeutung der Schnittstellen-Speicherzellen

Aus diesem Grund wird die Interrupt-Technik vom DOS, dem Betriebssystem der Floppy, intensiv genutzt. Vor allem werden alle Lese- und Schreiboperationen während der IRQ-Schleife ausgeführt. Dabei bilden die untersten Speicherzellen des RAM-Bereiches, die Adressen $0000 bis $0011 eine Schnittstelle zum Hauptprogramm der Floppy. Die Bedeutung der Speicherplätze zeigt Tabelle 2.

Durch Beschreiben einer der Speicherzellen $0000 bis $0005 läßt sich für den zugehörigen Puffer ein bestimmter Auftrag (Job) starten. Alle Befehle, die sich auf einen Block beziehen, benötigen zusätzlich noch die Track-/Sektor-Angabe in den entsprechenden Speicherzellen $0006 bis $0011. Die Aufträge mit ihren Codes zeigt Tabelle 3.

Wie Sie sehen, ist bei allen Jobcodes das höchstwertige Bit Nummer 7 gesetzt. Daran erkennt die Floppy, daß es sich um einen »Auftrag« handelt, und führt ihn während ihrer Interruptroutine aus. Nach der Abarbeitung hinterläßt sie eine Statusmeldung, deren höchstwertiges Bit immer gelöscht ist, und die eventuell aufgetretene Fehler signalisiert. Tabelle 4 zeigt die möglichen Statusmeldungen.

Ein Beispiel soll die Anwendung dieses Job-Prinzips verdeutlichen. Der Benutzerpuffer ist Puffer 2. Für ihn sind also die Job-Speicherzelle $0002 und der Track-/Sektorzeiger $000A/$000B zuständig. Um den Inhalt des Blocks 20, 1 in den Benutzerpuffer zu laden, benötigen wir damit ein Programm wie in Listing 1.

Datenaustausch zwischen Floppy und C 64

Glücklicherweise verfügt das Betriebssystem der Floppy über einige elementare Befehle. So ist es zum Beispiel möglich, vom Computer aus einzelne Bytes in den Speicher des Laufwerks zu schreiben. Diese Bytes können natürlich auch ein Programm bilden, das mit einem weiteren Kommando aufrufbar ist. Vielleicht sollten Sie sich die entsprechenden Befehle des 1541-DOS kurz noch einmal ins Gedächtnis rufen:

Memory-Write

Mit diesem Befehl läßt sich eine Anzahl von Bytes ins RAM der Floppy schreiben. Seine Syntax lautet folgendermaßen:

```
M-W CHR$ (LO) CHR$ (HI) CHR$ (Anzahl) CHR$
(Data 1)...
```

Dabei stehen »LO« und »HI« für niederwertigen beziehungsweise höherwertigen Teil der Floppy-Adresse und »Anzahl« für die Zahl der zu übertragenden Bytes. Allerdings ist diese Zahl auf maximal 34 begrenzt.

Memory-Execute

Für das Starten eines Programms im Floppy-RAM ist der Memory-Execute-Befehl zuständig. Syntax:

```
M-E CHR$ (LO) CHR$ (HI)
```

Das Programm wird während der Interrupt-Schleife des Floppy-Controllers ausgeführt und muß mit dem Befehl RTS abgeschlossen sein.

Bei Verwendung des Benutzerpuffers gibt es verkürzte Befehlsformen. So ist das Starten an der Adresse $0500 des Floppy-RAMs schlicht und einfach mit dem Befehl »UC« möglich.

Memory-Read

Selbstverständlich läßt sich der Floppy-Speicher auch lesen. Dies geschieht mit dem Memory-Read-Befehl. Hier wird die gewünschte Adresse mit:

```
M-R CHR$ (LO) CHR$ (HI)
```

übertragen, und deren Inhalt anschließend mit der Betriebssystem-Routine INPUT ($FFCF) abgeholt. Allerdings muß vorher der Eingabekanal mit CHKIN ($FFC6) festgelegt worden sein. Wir werden noch dazu kommen.

Daneben gibt es noch einige spezielle Befehle.

So wird mit dem Block-Execute-Befehl zunächst ein ganz bestimmter Block von der Diskette geholt, und dieser dann ausgeführt. Leider erscheint ein solches Programm nicht im Inhaltsverzeichnis und muß immer auf ein und demselben Block stehen. Kopieren auf eine andere Diskette ist also fast unmöglich.

Kaum bekannt ist auch der »&«-Befehl, da er in keinem offiziellen Handbuch auftaucht. Er ermöglicht das Laden von Daten in den Floppy-Speicher über einen Filenamen. Sein

```
START   LDA #20     ; Track-Nummer
        STA $0A     ; als Track für Puffer 2
        LDA #0      ; Sektor-Nummer
        STA $0B     ; als Sektor für Puffer 2
        LDA #$80    ; Jobcode für »Block lesen«
        STA $02     ; als Auftrag für Puffer 2
WARTEN  LDA $02     ; Status Puffer 2 holen
        BMI WARTEN  ; und warten bis Bit 7
                                  gesetzt
                    ;
        CMP #$01     ; Fehler aufgetreten ?
        BNE FEHLER  ; falls ja zur
                              Fehlerbehandlung
ENDE    RTS         ; sonst Ende
```

Listing 1. Ein kurzes, im Floppy-RAM ausführbares Programm

Jobcode	Funktion
$80	Lesen eines Blocks
$90	Schreiben eines Blocks
$A0	Verify eines Blocks
$B0	Suchen eines Blocks
$C0	Anschlagen des Kopfes (BUMP)
$D0	Programm im Puffer ausführen
$E0	Programm im Puffer ausführen, nachdem das Laufwerk hochgefahren wurde

Tabelle 3. Die Jobcodes des DOS der Floppy 1541 und ihre Funktionen

Format ist jedoch so unübersichtlich, daß die Erzeugung eines »&«-Files ein Programm für sich wert ist...

Genauere Angaben zu den genannten Befehlen findet man in dem Buch »Die Floppy 1541« von Karsten Schramm.

Programm über Bo(a)rd?

Mit dem obigen Assembler-Programm (Listing 1) liegt bereits der Quelltext eines kurzen Floppy-Programms vor. Wir können ihn in den Computer eintippen und von einem üblichen Assembler übersetzen lassen; er ist aber nur in der Floppy lauffähig. Für jeden Testlauf müßte man nun eine umständliche Prozedur auf sich nehmen:

Der assemblierte Quelltext, also der Objektcode, wird zunächst auf Diskette abgelegt, dann mit einem Monitor an eine andere Stelle im Computerspeicher geladen und in DATA-Zeilen verwandelt. Mit einem Basic-Ladeprogramm kann anschließend in einer Schleife jeder Wert einzeln zur Floppy übertragen werden.

Ein umständlicher und zudem auch noch sehr langsamer Vorgang, der einem die Arbeit schnell verleiden kann.

Mit dem Programm »Floppy-Lader« (Listing 2) bekommen Sie ein Hilfsmittel an die Hand, das schnell und so direkt wie möglich arbeitet. Genau wie beim Computer LOAD "Name",8,1 überträgt die kurze Routine ein Programm mit beliebigem Namen von der Diskette in den Speicher. Dabei wird auch hier die abgelegte Anfangsadresse berücksich-

Status	Bedeutung
$01	fehlerfreie Durchfühung
$02	Blockheader nicht gefunden
$03	SYNC nicht gefunden
$04	Datenblock nicht gefunden
$05	Datenprüfsumme falsch
$07	Fehler bei Verify
$08	Diskette schreibgeschützt
$09	Headerprüfsumme falsch
$0A	Datenblock zu lang
$0B	ID im Blockheader falsch
$0F	keine Diskette im Laufwerk
$10	Fehler bei Dekodierung

Tabelle 4. Statusmeldungen der Floppy

tigt. Will man jetzt ein Programm testen, braucht man es nur noch – wie gewöhnlich – für den gewünschten Bereich auf Diskette assemblieren zu lassen. Mit der Hilfsroutine wird der generierte Objektcode nach Angabe des Filenamens direkt an die richtige Stelle im Floppy-Speicher geladen. Der Start erfolgt dann mit einem Memory-Execute-Befehl. Auch die Länge des Datensatzes wird ganz genau wie beim Laden in den Computer abgelegt. Bild 1 zeigt das Format eines solchen Files auf Diskette. In jedem Block stellen die ersten beiden Bytes einen Zeiger auf den nächsten zugehörigen Block dar. Der letzte Block wird durch eine 0 als Nummer des Folgetracks gekennzeichnet. Die zweite Speicherstelle enthält dann die Anzahl der noch zum Eintrag gehörigen Bytes. Im ersten Block bilden zusätzlich die Bytes 2 und 3 die Anfangsadresse im Speicher. Erst Byte 4 enthält den ersten zu übertragenden Datenwert. Mit diesem Konzept wird auch Overlay-Technik möglich; das heißt, wenn in einem bestehenden Programm beispielsweise zehn Byte geändert werden sollen, können genau diese zehn Byte an die entsprechende Stelle geladen werden, ähnlich wie dies auch im Computer möglich ist. Man bleibt nicht an die gesamte Blocklänge von 256 Bytes gebunden.

Transporthilfe haarklein

Listing 2 zeigt den BASIC-Lader, der nach dem Abtippen mit RUN gestartet wird. Er enthält in den DATA-Zeilen das in der Floppy laufende Maschinenprogramm, das zunächst mittels einer FOR-NEXT-Schleife über den seriellen Bus zum Laufwerk geschickt wird. Per INPUT-Anweisung erhält der Anwender Gelegenheit, den Namen des nachzuladenden Files anzugeben. Abkürzungen mit »*« und »?«, den bekannten »Wild Cards«, sind wie gewohnt möglich. Nach RETURN wird der angegebene Filename mit Shift-Space (Dezimalwert 160) zu einer Länge von 16 Zeichen aufgefüllt und zur Floppy geschickt. Anschließend erfolgt der Start der Hilfsroutine und der Computer meldet sich sofort mit READY und wartet auf neue Befehle. Alles weitere erledigt die Diskettenstation von selbst. Sie arbeitet nun die übermittelte Maschinenroutine ab: Die gewünschten Daten werden in den Speicher der Floppy übertragen. Sollte der angegebene Filename auf der Diskette nicht vorhanden sein, blinkt die rote LED und die Fehlermeldung »FILE NOT FOUND« wird bereitgestellt. Gelöschte Files werden dabei vom Programm nicht berücksichtigt.

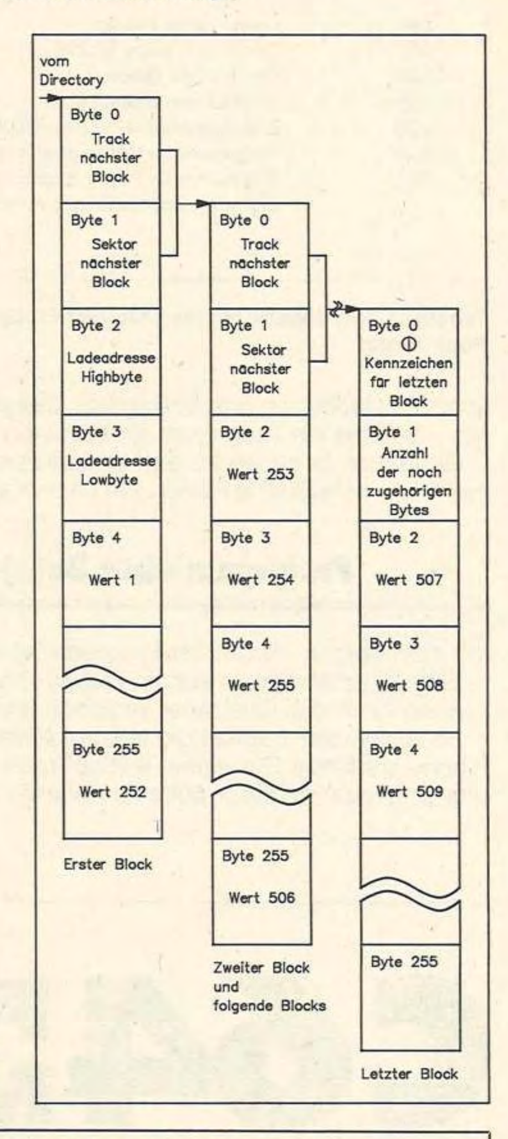


Bild 1. Aufbau eines Datenblocks auf Diskette

```
10 REM ***************************           <119>
20 REM *                         *           <069>
30 REM *      FLOPPY - LADER     *           <237>
40 REM *                         *           <089>
50 REM *      VON UWE GERLACH    *           <175>
60 REM *      RIEDESELSTR. 64    *           <158>
70 REM *      6100  DARMSTADT    *           <212>
80 REM *                         *           <129>
90 REM ***************************           <199>
100 :                                        <076>
110 REM  LAEDT EIN NORMALES FILE AB          <093>
120 REM  DER DORT ANGEGEBENEN ADRESSE        <005>
130 REM  IN DEN FLOPPY-SPEICHER.             <159>
140 REM  BELEGT PUFFER 1 IN DER FLOPPY       <124>
150 REM  ($0400 BIS $04FF).                  <189>
160 :                                        <136>
200 OPEN 1,8,15: REM BEFEHLSKANAL FLOPPY     <114>
400 PRINT: PRINT "MOMENT BITTE": PRINT: PR
    INT                                      <241>
510 FOR I=0 TO 217: REM ALLE DATAS           <110>
520 READ D: S=S+D                            <224>
530 PRINT#1,"M-W";CHR$(I);CHR$(4);CHR$(1);
    CHR$(D): REM IN PUFFER 1                 <040>
540 NEXT I                                   <116>
550 IF S<>21797 THEN PRINT"FALSCH! NUR MUT
    - NOCHEINMAL TIPPEN": END                <051>
700 INPUT "FILENAME";N$                      <134>
710 FOR I=1 TO 16: REM AUFFUELLEN            <146>
720 N$=N$+CHR$(160): REM SHIFT SPACE         <243>
730 NEXT I                                   <050>
740 N$=LEFT$(N$,16)                          <058>
750 PRINT#1,"M-W";CHR$(240);CHR$(4);CHR$(1
    6);N$: REM FILENAME NACH $04F0           <108>
800 PRINT#1,"M-E";CHR$(0);CHR$(4): REM HIL
    SPROGRAMM IM PUFFER 1 STARTEN            <182>
850 CLOSE 1: REM BEFEHLSKANAL SCHLIESSEN     <082>
900 :                                        <114>
901 DATA 173,0,28,9,8,141,0,28,32,29,4,173
    ,0,28,41,247,141,0,28,138,240,6          <153>
902 DATA 32,199,230,32,44,193,96,32,177,4,
    160,2,132,55,162,0,185,0,3,240,44        <090>
903 DATA 200,200,200,189,240,4,201,42,240,
    15,201,63,240,5,217,0,3,208,25,200       <172>
904 DATA 232,224,16,208,234,164,55,200,185
    ,0,3,141,178,4,200,185,0,3,141,182       <112>
905 DATA 4,76,117,4,165,55,24,105,32,168,1
    32,55,144,195,173,0,3,208,3,162          <069>
906 DATA 57,96,141,178,4,173,1,3,141,182,4
    ,76,29,4,32,177,4,169,0,133,167          <008>
907 DATA 173,3,3,133,168,172,2,3,162,4,169
    ,0,133,53,173,0,3,208,12,173,1,3         <062>
908 DATA 24,105,1,133,53,32,194,4,96,141,1
    78,4,173,1,3,141,182,4,32,194,4          <083>
909 DATA 208,241,32,177,4,162,2,76,134,4,1
    69,18,133,6,169,1,133,7,169,128          <009>
910 DATA 133,0,165,0,48,252,96,189,0,3,145
    ,167,200,208,4,230,168,240,8,232         <165>
911 DATA 228,53,208,239,162,0,96,162,82,96
    ,200                                     <205>
```

Listing 2. Der »Floppy-Lader« überträgt ein Programm ins Floppy-RAM

Listing 3 zeigt den ProfiAss-Quelltext der Hilfsroutine. Er braucht selbstverständlich nicht abgetippt zu werden, sondern dient lediglich der Dokumentation. Wichtig ist er auch zur erneuten Assemblierung für einen anderen Teil des Floppy-RAMs.

Man erkennt, daß die Hilfsroutine selbst eigentlich ein großes Unterprogramm darstellt. Es wird eingeschlossen von einer übergeordneten Ebene, die lediglich die LED-Steuerung und Fehlerbehandlung übernimmt.

Die erste Hauptaufgabe besteht in der Suche nach dem gewünschten Directory-Eintrag, der ja auch Spur und Sektor des ersten File-Blocks enthält. Bild 2 veranschaulicht den Aufbau der Directory-Blöcke. Sie enthalten jeweils acht File-Einträge. Jeder Eintrag hat wiederum das gleiche Format. Es wird auf der rechten Seite von Bild 2 dargestellt. Wichtig für unsere Routine sind lediglich der Filetyp, der Filename und natürlich Anfangstrack und -sektor.

Um den richtigen Eintrag zu finden, wird von Block 18, 1 aus die gesamte Directory-Liste folgendermaßen durchsucht: Das Unterprogramm »LIES« holt einen Block von der Diskette in den Hauptarbeitspuffer 0 ($0300 bis $03FF). Das Y-Register dient als Zeiger in diesen Puffer und wird

```
10 rem open 1,8,1,"0:floppy-lader.obj"
12 open 2,4
13 sys 32768
14 ;
15 ; **********************
16 ; *    floppy-lader    *
17 ; *                    *
18 ; *  von  uwe gerlach  *
19 ; * tel.  06151/316205 *
20 ; *   bzw 06622/3908   *
21 ; *                    *
22 ; *      06.04.86      *
23 ; **********************
40 ;
50 ; laedt ein file ins floppy-ram
60 ; ab der gespeicherten adresse.
70 ; filename steht ab name im floppy-
80 ; speicher.
90 ;
100 .tit "floppy-lader c64 & vc1541    "
120 .opt p2
125 ;
130 *=$0400; assemblieren fuer die floppy
150 ;
200 zeiger = $37
210 laenge = $35
220 name = $04f0;anfang des filenamens
230 led = $1c00;adresse fuer led (bit 3)
1000 ;
1010 ; led bedienung und programmaufruf
1020 ;
1050 lda led
1060 ora #$08
1070 sta led;rote led einschalten
1080 ;
1090 jsr suche;dienstprogramm aufrufen
1100 ;
1120 lda led
1130 and #%11110111
1140 sta led;rote led ausschalten
1150 txa;fehlernummer in den akku
1160 beq ok;falls kein fehler aufgetreten
1180 ;
1190 jsr $e6c7;fehlermeldung bereitstellen
1200 jsr $c12c;led blinken
1210 ok rts;ende
2800 ;
2900 ;
2910 ; anfangsblock des files suchen
2920 ;
3000 suche jsr lies;block 18,1 nach puffer 0
3010 ldy #2;anfang erster fileeintrag
3020 sty zeiger
3030 eintr ldx #0
3040 lda $0300,y;filetyp lesen
3050 beq nix;ueberspringen, falls del
3060 iny:iny
3070 iny;zeiger y auf filenamen
3080 pruef lda name,x;zeichen aus filename holen
3090 cmp #"*";auf joker pruefen
3100 beq found;aktuellen fileeintrag uebernehmen
3110 cmp #"?";auf wild card pruefen
3120 beq wild;weiter mit naechstem zeichen
3130 cmp $0300,y;mit directoryzeichen vergleichen
3140 bne nix;falls ungleich naechster fileeintrag
3150 wild iny
3160 inx
3170 cpx #16;laenge des filenamens
3180 bne pruef;falls noch zeichen uebrig
4000 ;
4010 ; filename gefunden
4020 ;
4050 found ldy zeiger
4070 iny;zeiger y auf track/sektor herstellen
4080 lda $0300,y;anfangstrack holen
4090 sta lies+1;und sichern
4100 iny
4110 lda $0300,y;anfangssektor holen
4120 sta sektor+1;und sichern
4130 jmp laden
5000 ;
5010 ; falls noch nicht gefunden
5020 ;
5050 nix lda zeiger
5060 clc;addition vorbereiten
5070 adc #32;anfang naechster eintrag berechnen
5080 tay;als zeiger sichern
5090 sty zeiger;und zwischenspeichern
5100 bcc eintr;naechster block, falls vorhanden
5110 lda $0300;trackangabe naechster block
5120 bne neubl;falls nicht 0 und damit dir-ende
5130 ldx #$39;fehlernummer "file not found"
5138 rts;ende, da file nicht vorhanden
5140 neubl sta lies+1;als track fuer lesebefehl
5150 lda $0301;sektorangabe naechster block
5160 sta sektor+1;als sektor fuer lesebefehl
5170 jmp suche;weiter mit naechstem block
6000 ;
6010 ; file in den floppy-speicher laden
6020 ;
6040 laden jsr lies;ersten fileblock in den puffer
6050 lda #0
6060 sta $a7;kopierzeiger lowbyte
6070 lda $0303;anfangsadresse high
6080 sta $a8;kopierzeiger highbyte
6090 ldy $0302;anfangsadresse low
6100 ldx #4;erstes gueltiges byte
6110 nextbl lda #0;gesamtlaenge eines vollen blocks + 1
6120 sta laenge;erstes unbelegtes byte
6140 lda $0300;erstes byte (folgetrack)
6150 bne next;falls noch weitere bloecke
6160 ende lda $0301;zweites byte (anzahl bytes)
6170 clc;addition vorbereiten
6179 adc #1
6180 sta laenge;erstes unbelegtes byte
6190 jsr trans;rest kopieren
6195 rueck rts;zurueck zur led-kontrolle
7000 next sta lies+1;track naechster block
7010 lda $0301;zweites byte (folgesektor)
7020 sta sektor+1;sektor naechster block
7030 jsr trans;block in speicher kopieren
7035 bne rueck;fehler beim kopieren
7040 jsr lies;naechsten block einlesen
7050 ldx #2;erstes gueltiges byte
7060 jmp nextbl;weiter mit naechstem block
8000 ;
8010 ; unterroutinen
8020 ;
8050 lies lda #18
8060 sta $06;als track fuer puffer 0
8070 sektor lda #1
8080 sta $07;als sektor fuer puffer 0
8090 lda #$80;jobcode fuer "block lesen"
8100 sta $00;als auftrag fuer puffer 0
8110 weiter lda $00;status puffer 0
8120 bmi weiter;falls noch nicht fertig
8130 rts
9000 ;
9050 trans lda $0300,x;byte aus puffer
9060 sta ($a7),y;in den floppy-speicher
9070 iny;ram-zeiger erhoehen
9080 bne cont;falls kein low-byte ueberlauf
9090 inc $a8;high-byte erhoehen
9095 beq over;zeropage schuetzen
9100 cont inx;pufferzeiger erhoehen
9110 cpx laenge;schon fertig
9120 bne trans;weitermachen, falls nicht
9130 ldx #0;fehlerfreier ablauf
9150 rts;falls puffer kopiert
9200 ;
9210 over ldx #$52;fehlernummer "file too large"
9220 rts
```

Listing 3. Der Top-Ass-Quelltext des Assemblerteils im »Floppy-Lader«

zunächst auf 2 gesetzt, weist also auf die Speicherstelle, die den Filetyp des ersten Eintrags angibt. Falls ihr Inhalt 0 ist, was den Filetyp »Deleted« kennzeichnet, wird gleich mit dem nächsten Eintrag fortgefahren. Ansonsten setzt dreimaliges INY den Zeiger gleich auf den Anfang des Filenamens. Er wird unter Berücksichtigung der Wild-Cards Byte für Byte mit dem vorgegebenen Namen verglichen, wobei das X-Register einen Zeiger in den Vergleichsnamen bildet.

Beim ersten falschen Zeichen wird die Suche bereits mit dem nächsten Eintrag fortgesetzt. Dies geschieht durch Addition von 32 zur in »ZEIGER« zwischengespeicherten Basisadresse des vorhergehenden Eintrags. Erfolgt ein Überlauf, so befindet sich der nächste Eintrag im folgenden Block, der in diesem Fall zunächst geladen werden muß.

War die Suche erfolgreich, werden Track und Sektor des ersten Fileblocks isoliert und es erfolgt der Sprung zum zweiten Hauptteil des Hilfsprogramms, der Laderoutine.

Nach dem Laden des ersten Blocks – ebenfalls wieder in den Hauptarbeitspuffer 0 ($0300 bis $03FF) – werden zunächst Low- und Highbyte der Ladeadresse isoliert. Dann folgen Einstellung der Parameter und Aufruf der Kopierroutine »TRANS«, die alle gültigen Bytes aus dem Puffer in den entsprechenden RAM-Teil überträgt. Die Speicherstellen $A7 und $A8 in der Zeropage bilden dabei zusammen mit dem Y-Register einen Zeiger in den Zielbereich, das X-Register einen Zeiger in den Puffer.

Anschließend lädt das Programm sofort den Folgeblock nach. Hieraus werden nur Folgetrack und -sektor isoliert und dann wieder die gültigen Bytes kopiert.

Das geht so weiter, bis die Tracknummer 0 das Fileende signalisiert. Aus dem letzten Block werden jetzt nur die signifikanten Bytes berücksichtigt.

Beim Rücksprung zur obersten Ebene des Hilfsprogramms muß die hexadezimale Fehlernummer im X-Register stehen. Fehlerfreien Ablauf kennzeichnet eine 0. In diesem Fall wird die rote LED ausgeschaltet und es erfolgt der Rücksprung. Steht aber ein anderer Wert im X-Register, dann wird zusätzlich die DOS-Routine $E6C7 aufgerufen, um den entsprechenden Fehlertext in der Floppy bereitzustellen. Das Unterprogramm $C12C veranlaßt das Blinken der LED.

Der serielle Bus

Die 1541 ist am seriellen Bus des C64 angeschlossen. Leider wurde bei der Konzeption dieses wichtigen Datenkanals tüchtig gespart. Übrig blieb eine abgemagerte Version des parallelen IEC- oder auch IEEE-488-Busses, mit dem die Geräte der CBM-Serie von Commodore ausgerüstet sind. Die Kommunikation mit den angeschlossenen Geräten ähnelt zwar dem parallelen Vorbild und sogar die Bedienung von Maschinensprache aus geschieht über Routinen, die von der Funktion her identisch mit denen der großen CBM-Floppies sind, doch können diese Tatsachen kaum darüber hinwegtrösten, daß beim seriellen Konzept nur eine einzige Datenleitung zur Verfügung steht, was alle Übertragungen zwangsweise verlangsamen muß.

Im folgenden sollen die Vorgänge auf dem seriellen Bus kurz vorgestellt werden. Zur Bedienung können Sie universelle Betriebssystem-Routinen einsetzen, die im Anschluß erläutert werden. Trotzdem sollen Sie ein Gefühl dafür bekommen, was mit jedem einzelnen Byte so alles vor sich geht, bis es schließlich auf der anderen Seite angekommen ist. Vielleicht lassen Sie sich dadurch auch inspirieren, einfachere und schnellere Routinen selbst zu schreiben.

Fangen wir mit der Hardware an. Tabelle 5 zeigt, wie die einzelnen Leitungen sowohl im C64 als auch in der Floppy gesteuert und abgefragt werden können. Dabei ist zu beachten, daß getrennte Anschlüsse für Empfangen und Senden existieren. In die Sendeleitungen sind bei beiden Geräten Inverter mit Open-Collektor-Ausgängen vom Typ 7406 geschaltet. In der Floppy enthalten auch die Empfangsleitungen invertierende Schmitt-Trigger vom Typ 74LS14. Dagegen sind die bezeichneten Eingänge des Computers direkt an die Busleitungen gekoppelt.

Es gibt drei Operationsarten auf dem seriellen Bus:
1. CONTROL (steuern)
2. TALK (senden)
3. LISTEN (empfangen)

Der Controller koordiniert die Aktionen auf dem Bus, ein Talker beeinflußt die Datenleitung und ein Listener empfängt Informationen.

Der C64 arbeitet sowohl als Talker, wenn er zum Beispiel Daten zum Drucker überträgt, wie auch als Listener, wenn er beispielsweise ein Programm von Diskette lädt. Andere Geräte können entweder Listener (Drucker), Talker (Meßge-

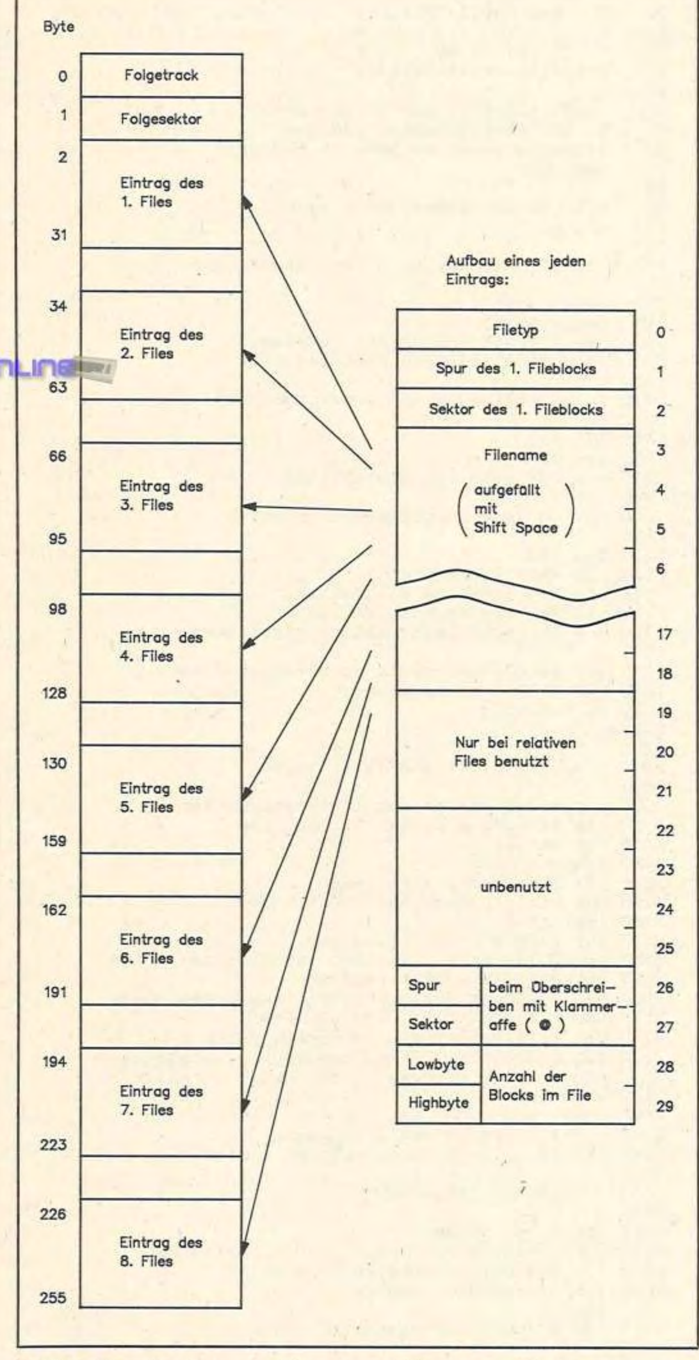


Bild 2. Der Aufbau der Directory-Blöcke auf Diskette

rät) oder beides sein (Floppy). Auf dem Bus kann immer nur ein Talker arbeiten, jedoch können durchaus mehrere Listener aktiv sein.

Damit Daten zu einem bestimmten Empfänger übertragen werden können, erhält jedes Gerät eine Nummer, also eine Adresse. Beim C64 hat Commodore für den seriellen Bus die Gerätenummern 4 bis 31 reserviert.

Der Ablauf einer Busübertragung

Über den seriellen Bus kann der C64 jeweils ein bestimmtes der angeschlossenen Geräte auffordern zu empfangen oder zu senden. Das tut er, indem er die Leitung ATN (ATteNtion = Achtung) auf LOW zieht. Alle angeschlossenen Geräte unterbrechen daraufhin ihre augenblickliche Tätigkeit und gehen in den empfangsbereiten Zustand. Als nächstes legt der Controller nun Bit für Bit die Geräteadresse auf den Bus - im Falle der Diskettenstation also beispielsweise die 8 - während ATN auf LOW bleibt. Alle nicht adressierten Geräte ziehen sich danach vom Busbetrieb zurück und setzen ihre unterbrochene Arbeit fort. Das angesprochene Gerät aber muß innerhalb einer Millisekunde reagieren, indem es die Datenleitung nach LOW zieht. Andernfalls nimmt der Computer an, das adressierte Gerät sei nicht am Bus und gibt eine entsprechende Fehlermeldung (DEVICE NOT PRESENT ERROR) aus. ATN wird zurückgesetzt und dem Gerät mit TALK bzw. LISTEN mitgeteilt, ob es Daten senden oder empfangen soll.

Ist die Übertragung beendet, so wird das Gerät wieder deadressiert. Dazu sendet der Computer entweder UNTALK, wenn das Gerät vorher Sender war, oder UNLISTEN, sofern es sich im Empfangsmodus befand.

Sollen zum Beispiel Daten von der Floppy geholt werden, sind nach Geräteadresse und LISTEN den Filenamen und alle wichtigen Fileparameter anzugeben. Nach UNLISTEN kann der Computer in beliebigen Zeitabständen Daten von der Floppy empfangen. Er legt dazu — wie oben — die Leitung ATN auf LOW, gibt danach die Gerätenummer an und schickt ein TALK über den Bus. Daraufhin stellt die Floppy solange Daten bereit, bis vom Computer ein UNTALK-Signal kommt. Auch die Floppy kann ihrerseits die Übertragung beenden (zum Beispiel nachdem das letzte Programmbyte übermittelt wurde), indem sie ein EOI (End Of Information = Informationsende) aussendet. Der C64 schließt daraufhin mit einem UNTALK die Übertragung ab.

Da vor jeder Übertragung die Gerätenummer angegeben werden muß, ist es dem Controller möglich, gleichzeitig mehrere Files auf dem Bus offenzuhalten, um völlig gemischt auf die einzelnen Geräte zugreifen zu können.

Damit die Datenübertragung richtig funktioniert, muß sichergestellt werden, daß ein Folge-Byte erst dann übermittelt wird, wenn der Empfänger das vorhergehende bereits verarbeitet hat. Dazu gibt es sogenannte Handshake-Signale. Der Talker teilt durch Freigeben der CLK-Leitung mit, daß er Daten senden will. Sobald der Listener bereit ist, gibt er seinerseits die DATA-Leitung frei, und der Talker beginnt daraufhin mit der Übertragung eines Bytes. Dabei wechselt der Zustand der Datenleitung während der negativen CLK-Phase und wird immer beim LOW-/HIGH-Übergang des Taktes übernommen. Der Listener zieht nach dem vollständigen Empfang eines Bytes die DATA-Leitung solange nach LOW, bis er wieder empfangsbereit ist und das nächste Byte folgen kann.

Obwohl sich die Beschreibung dieses Vorgangs kompliziert anhört, kann nur so eine sichere Übertragung mit der geforderten Universalität und der unter den gegebenen Umständen maximal erreichbaren Geschwindigkeit vonstatten gehen.

Dank der im Betriebssystem des C64 vorhandenen universellen Bus-Routinen vereinfacht sich die Datenübertragung zur Floppy ungemein. Die nachfolgend aufgeführten Funktionen lassen sich ähnlich wie die entsprechenden BASIC-Befehle anwenden, wenn man ihre Parameter und Einsprungbedingungen kennt.

OPEN ($FFCO)
Die Routine »OPEN« öffnet einen Kanal. Vorher müssen mit der Routine SETFLS die logische Filenummer, die Gerätenummer und die Sekundäradresse sowie mit SETNAM der Filename festgelegt sein.

SETFLS ($FFBA)
legt die Fileparameter fest. Vor dem Aufruf müssen die Register folgende Infomationen enthalten:
Akku - logische Filenummer
X - Gerätenummer
Y - Sekundäradresse ($FF falls keine Sekundäradresse)

SETNAM ($FFBD)
legt einen Filenamen fest. Die Register haben folgende Bedeutung:
Akku - Länge des Namens (0 falls kein Name)
X - Lowbyte der Anfangsadresse des Namens
Y - Highbyte der Anfangsadresse des Namens

PRINT ($FFD2)
gibt den Wert im Akku auf das mit CHKOUT festgelegte Gerät aus. Ohne Aufruf von CHKOUT oder nach einem Fehler geht die Ausgabe auf den Bildschirm.

Adresse	C64	Bit	Bus-Signal	Richtung	VC1541 Adresse		Bit
$DD00	56576	3	ATN	→	$1800	6144	7
$DD00	56576	5	DATA	→	$1800	6144	0
$DD00	56576	7		←	$1800	6144	1
$DD00	56576	4	CLK	→	$1800	6144	2
$DD00	56576	6		←	$1800	6144	3
$DC0D	56589	4	SRQ	←	—	—	—

Tabelle 5. Die Beschaltung des seriellen Busses, der Daten zwischen C64 und Floppy überträgt

CHKOUT ($FFC9)
leitet die Ausgabe der Routine PRINT auf den Kanal mit der im X-Register festgelegten logischen Filenummer um. Der Kanal muß vorher mit OPEN geöffnet worden sein. Das entspricht der Wirkung des BASIC-Befehls CMD.

INPUT ($FFCF)
holt ein Zeichen von dem mit CHKIN festgelegten Gerät und übergibt es im Akku. Ohne Aufruf von CHKIN erfolgt die Eingabe von der Tastatur.

CHKIN ($FFC6)
leitet die Eingabe der Routine INPUT auf den Kanal mit der im X-Register festgelegten logischen Filenummer um.

CLRCH ($FFCC)
hebt die Wirkung von CHKOUT und CHKIN wieder auf. Ausgaben erfolgen danach auf den Bildschirm, Eingaben von der Tastatur.

CLOSE ($FFC3)
schließt die Datei, deren logische Filenummer im Akku steht. Der Kanal muß vorher mit OPEN geöffnet werden.

Bei der Übermittlung von Befehlen an die Floppy müssen einige Besonderheiten berücksichtigt werden:

Wie auch unter Basic dient zur Übermittlung der Kommandokanal mit der Sekundäradresse 15. Die Befehle werden über ihn Zeichen für Zeichen mit der Routine PRINT ausgegeben. Als Abschluß muß mit CLRCH die Ausgabe wieder auf den Bildschirm geleitet werden. Erst dann erkennt die Floppy, daß die Sequenz abgeschlossen ist und führt den erhaltenen Auftrag aus.

Ein Kopierprogramm

Wie ein Maschinenprogramm via Basic zur Floppy transportiert und gestartet werden kann, wissen Sie spätestens seit Listing 2. Natürlich ist es gerade bei der Maschinenprogrammierung des C64 sinnvoll, schnell einmal Aufgaben an die Floppy delegieren zu können. Hier soll nun eine universelle Maschinenroutine für den C 64 vorgestellt werden, die entsprechende Programme in den Benutzerpuffer (ab $0500) der Floppy schreibt und dort startet (Listing 4).

```
         LDA **#1       ; logische Filenummer
         LDX **#8       ; Gerätenummer der
                          Floppy
         LDY **#15      ; Sekundäradresse
         JSR SETFLS     ; als Parameter
                          festlegen
         LDA **#00
         JSR SETNAM     ; kein Name
         JSR OPEN       ; OPEN 1,8,15
         LDY **#00
         STY ADR        ; Kopier Ziel-
                          Basisadresse
MWBEFEHL LDX **#01      ; Filenummer
                          Befehlskanal
         JSR CHKOUT     ; Ausgaben zur Floppy
                          leiten
         LDA **# "M"
         JSR PRINT
         LDA **#"-"
         JSR PRINT
         LDA **#"W"
         JSR PRINT      ; "M-W...
         LDA ADR        ; Zieladresse Lowbyte
         JSR PRINT
         LDA **#5       ; Zieladresse Highbyte
         JSR PRINT
         LDA **#34      ; Anzahl der folgenden Byte
         JSR PRINT
         LDX **#0       ; Zähler für
                          Befehlslänge
FUELLEN  LDA START,Y    ; Byte der Floppy-Routine
                          holen
         JSR PRINT      ; und in den Befehlspuffer
         INY            ; Offset erhöhen
         CPY **#ENDE-START ; War's das ?
         BEQ COPYOK     ; falls ja, Kopiervorgang
                          beenden
         INX            ; sonst Befehlslänge
         CPX **#34      ; auf Maximum überprüfen
         BNE FUELLEN    ; und falls möglich
                          auffüllen
         ADC **#34      ; aktualisieren
         STA FNAM       ; zurückschreiben
         JMP MWBEFEHL; und neuen M-W-Befehl
                          generieren
```

```
         JSR CLRCH      ; andernfalls Befehl
                          abschicken
         LDA ADR        ; alte Zieladresse holen
         CLC            ; Addition vorbereiten
COPYOK   JSR CLRCH      ; Befehl ausführen
         LDX **#1       ; logische Filenummer
         JSR CHKOUT     ; Ausgabe auf Kommandokanal
         LDA **#"U"
         JSR PRINT
         LDA **#"C"
         JSR PRINT      ; "UC" (Floppy-Programm
                          starten)
         JSR CLRCH      ; ausführen und normale
                          Ausgabe
         LDA **#1       ; logische Filenummer
         JSR CLOSE      ; Kommandokanal schließen
```

Listing 4. Kopier-Routine für Floppy-Programme als Beispiel für Floppy-Programmierung in Maschinensprache

Das zu kopierende Programm darf nicht länger als 255 Byte sein, muß also in einen einzigen Puffer passen. Außerdem muß es mit dem Label START beginnen und mit ENDE aufhören. Falls es komplett relokativ geschrieben wurde, läßt es sich sogar in einem Zug mitassemblieren, da sein Code dann unabhängig vom Adreßbereich lauffähig ist. In diesem Fall darf es nur relative Sprünge (Branches) enthalten und keine eigenen Teile als Unterprogramm aufrufen.

Die Gegenseite

Mindestens genauso wichtig wie das Übermitteln von Daten an die Floppy, ist aber das Abholen von Ergebnissen. Das folgende Programm (Listing 5) ist ein Beispiel zum Lesen der Daten aus dem Benutzerpuffer. Zuerst wird auch hier wieder ein Befehlskanal zur Floppy geöffnet, über den anschließend mittels des Memory-Read-Kommandos Zeichen für Zeichen gelesen und angezeigt wird.

```
         LDA **#1       ; logische Filenummer
         LDX **#8       ; Gerätenummer der Floppy
         LDY **#15      ; Sekundäradresse
         JSR SETFLS     ; als Parameter festlegen
         LDA **#0
         JSR SETNAM     ; kein Name
         JSR OPEN       ; OPEN 1,8,15
         LDX **#1       ; Filenummer Befehlskanal
         JSR CHKOUT     ; Ausgaben zur Floppy
         LDX **#0       ; Zähler initialisieren
MRBEFEHL LDA MR,X       ; Text "M-R 00 05 00"
         JSR PRINT      ; ausgeben
         INX
         CPX **#MW-MR   ; schon fertig ?
         BNE MRBEFEHL; weiter, falls nicht
         JSR CLRCH      ; Ausgabe wieder normal
                        ;
         LDX **#1       ; Filenummer Befehlskanal
         JSR CHKIN      ; Eingaben von der Floppy
ZEICHEN  JSR INPUT      ; ein Zeichen von dort holen
         JSR PRINT      ; und anzeigen
         LDA $90        ; Statusvariable des C 64
                          lesen
```

Listing 5. Auslesen eines Floppy-Puffers vom C64 aus

```
        CMP **#64       ; und auf EOI testen
       BNE ZEICHEN ; weiter lesen, falls kein EOI
        LDA **#1        ; logische Filenummer
        JSR CLOSE     ; Kommandokanal schließen
        RTS           ; sonst Ende
        MR        .ASC "M-R"
        .BYTE $00, $05, $00, $0D
MW      .END
```

Der Text für den Memory-Read-Befehl ist am Anschluß an das Programm abgelegt. Das Ende der Übertragung wird automatisch über das EOI (End Of Information) Bit des C64-Statusregisters angezeigt.

Greift der C64 über den seriellen Bus auf die Floppy zu, während dort gerade ein Benutzerprogramm abläuft, dann wird der C64 zwangsweise angehalten, bis dieses Programm beendet ist. So ist gewährleistet, daß zum Beispiel ein Puffer nicht ausgelesen wird, bevor die gewünschten Daten bereitgestellt wurden.

Nach diesen Beispielen soll nun die Anwendung der vorgestellten Routinen gezeigt werden.

Wer kann sich den genauen Namen eines selten benötigten Programms schon merken? Der übliche Weg beim Laden ist daher die Auswahl aus dem Disketteninhaltsverzeichnis. Dazu muß zunächst mit LOAD "$",8 das Directory geholt und das Ergebnis mit LIST angezeigt werden. Nun läßt sich der Cursor an den Anfang der entsprechenden Zeile bewegen. LOAD wird vor den angezeigten Namen geschrieben, der Cursor wird nach rechts bewegt und das unvermeidliche ,8: oder lieber gleich ,8,1 dahintergesetzt. Jetzt noch <RETURN> eingeben und die Floppy legt los. Anschließend noch ein RUN und es geht los.

Alles in allem doch ein wenig umständlich, oder?

Problematisch wird es erst bei neuen oder fremden Disketten. Immer häufiger bestehen Programme aus mehreren einzelnen Files. Deren Namen erscheinen alle im Inhaltsverzeichnis, auch wenn sie selbst nicht ausführbar sind, sondern nur nachgeladen werden. Läßt man das Inhaltsverzeichnis einer gut gefüllten Diskette ausgeben, kann das mitunter schon sehr verwirrend sein. Wie soll man herausfinden, welches das jeweilige Startprogramm ist?

Zudem passen bald gar nicht mehr alle Filenamen auf den Bildschirm, so daß mehrmals das Directory gelistet werden muß.

Bei Maschinenprogrammen wird es noch unübersichtlicher. Mit nachgestelltem »,8,1« werden sie zwar (hoffentlich) automatisch an die richtige Stelle geladen, aber wie war doch gleich die Startadresse für den SYS-Befehl?

In solchen Fällen hilft ein Menü-Programm, das nur die wichtigen Namen enthält und auf Knopfdruck das entsprechende Boot-File lädt und startet. Nun ist es etwas müßig, für jede Diskette einzeln ein solches Menü-Pogramm zu entwerfen. Immerhin müßte die Grundstruktur doch überall etwa gleich aussehen.

Aus dieser Überlegung heraus entstand der »UNIVERSAL-LADER« (Listing 6). Startet man ihn, werden alle Einträge der Diskette als Liste auf dem Bildschirm angezeigt, sofern sie nicht – und das ist der Clou der Geschichte – mit dem zugehörigen Initialisierungsprogramm gekennzeichnet wurden. Dadurch bleibt das Menü übersichtlich.

Mit den Cursortasten läßt sich nun ein File auswählen und mit <RETURN> starten, egal, ob es sich um ein Basic- oder ein Maschinenprogramm handelt. Mit dem Initialisierungsprogramm läßt sich nämlich auch eine beliebige Startadresse für jedes einzelne File vorgeben. Zusätzlich besteht noch die Möglichkeit, Namen in die Liste aufzunehmen, die beim Auswahlvorgang nicht berücksichtigt werden, da der Cursor sie einfach überspringt. So läßt sich durch Trennstriche zwischen zusammengehörigen Filegruppen die Übersichtlichkeit noch erhöhen.

Der Universal-Lader sollte als erstes File auf jeder Diskette stehen. Statt mit LOAD "$",8 und LIST das Inhaltsverzeichnis zu holen, läßt er sich dann ebenso kurz mit LOAD "*",8 und RUN aktivieren. Manche alternativen Betriebssysteme, wie etwa Speeddos, bieten sogar eine Funktion zum Laden und Starten des ersten Programms mit einem einzigen Tastendruck (<SHIFT RUN/STOP>). Der Universal-Lader belegt mit 5 Blocks nicht übermäßig viel Platz. Außerdem bietet sich die Möglichkeit an, das kurze Programm in die Directory-Spur zu übernehmen, was auch Geschwindigkeitsvorteile bringt, da gleich nach dem Laden das Inhaltsverzeichnis gelesen werden muß. Somit entfällt der zeitraubende zweifache Spurwechsel.

Nach einem Diskettenwechsel bewirkt Drücken von <SPACE>, daß ein neues Menü erstellt wird.

Verwaltungssache

Den Aufbau des Inhaltsverzeichnisses auf jeder Diskette kennen Sie ja bereits. Wichtig für normale Programme sind der Filetyp (PRG, REL, USR, DEL...), der in Byte 0 jedes Eintrags verschlüsselt ist, die darauf folgenden Bytes 1 und 2, die einen Zeiger auf den Anfangsblock des jeweiligen Files bilden, und die Bytes 3 bis 18, die den Filenamen enthalten. Andere Speicherplätze haben Sonderaufgaben; die Bytes 22 bis 25 jedoch erfüllen im normalen Commodore-DOS keine Funktion (unter Geos werden diese Bytes jedoch verwendet), so daß sie für eigene Anwendungen zur Verfügung stehen. Nach dem Formatieren enthalten sie den Wert 0. Diese freien Bytes können nun gezielt mit dem Initialisierungsprogramm geändert werden. Ihre Funktion ist Tabelle 6 zu entnehmen.

Das Menü-Programm

Der Universal-Lader enthält mehrere programmiertechnische Besonderheiten, so daß es sich lohnt, die Beschrei-

Byte	Wert	Funktion
22	0	Name wird im Menü angezeigt
	1 bis127	Name wird angezeigt, aber übersprungen (Skip)
	128 bis 255	Name wird nicht angezeigt
23	0 bis 255	Lowbyte der Startadresse
24	0 bis 255	Highbyte der Startadresse

Tabelle 6. Funktion der vom Universal-Lader benutzten Directory-Bytes

Adresse	Inhalt	Bedeutung
$0801 = 2049	$0B	Adresse Anfang der
$0802 = 2050	$08	nächsten Basic-Zeile
$0803 = 2051	$0A	Zeilen-Nummer (10)
$0804 = 2052	$00	der Basic-Zeile
$0805 = 2053	$9E	Token für SYS
$0806 = 2054	$32	Zeichen »2«
$0807 = 2055	$30	Zeichen »0«
$0808 = 2056	$36	Zeichen »6«
$0809 = 2057	$31	Zeichen »1«
$080A = 2058	$00	
$080B = 2059	$00	Kennung Basic-Ende
$080C = 2060	$00	

Tabelle 7. Basic-Kopf des Programms »Universal-Lader«

bung zu verfolgen. Manche Kniffe lassen sich auch gut in eigene Programme einbauen.

Um eine möglichst kurze Ausführungszeit bei kompaktem Programmcode zu erzielen, wurde der Universal-Lader in Assembler geschrieben. Damit er trotzdem wie ein normales Basic-Programm geladen und mit RUN gestartet werden kann, erhielt er einen Basic-Kopf. Tabelle 7 zeigt die Bedeutung der einzelnen Bytes. Sie erzeugen am Basic-Start $0801 die Zeile

```
10 SYS 2061
```

Dezimal 2061 entspricht hexadezimal $080D, so daß der Einsprung unmittelbar an der ersten »freien« Adresse erfolgt. Die drei NOP-Befehle dort halten Platz frei für einen Sprungbefehl, von dem später noch die Rede sein wird.

Der grobe Aufbau des Assembler-Progamms sieht folgendermaßen aus:

Zunächst werden alle Dateinamen gelesen und im Speicher des C 64 ab Adresse $7000 abgelegt. Soviele Namen auf den Bildschirm passen, werden dort angezeigt, wobei eine Zeile farblich hervorgehoben ist. Die Namen lassen sich nun auf und ab scrollen. Mit <RETURN> wird der farblich hervorgehobene Name in der ganz normalen LOAD-Routine zur Floppy übermittelt, das zugehörige Programm geladen und gestartet.

Die langsame Datenübertragung zwischen Computer und Floppy macht sich nicht so stark bemerkbar, wenn weniger Informationen übertragen werden. Zum Glück ist es auch gar nicht nötig, das gesamte Inhaltsverzeichnis zu übermitteln. Der Universal-Lader benötigt ohnehin nur die Filenamen und die Startadressen der Programme, die tatsächlich im Menü erscheinen sollen. Die Selektierung geschieht mit einem Programmteil, der in der Floppy selbst läuft. Er ist voll relokativ geschrieben, benutzt also keine direkten Sprünge, und bleibt daher unverändert auch in einem anderen Adreßbereich lauffähig. Eine kurze Routine schiebt diesen Programmteil in den Floppy-Speicher (ab Adresse $0500) und startet ihn dort. Als Datenpuffer wird der Speicherbereich $0400 bis $0500 in der Floppy benutzt.

Schneller durch Arbeitsteilung

Soweit möglich, wird der Vorteil, mit zwei Prozessoren zu arbeiten, in diesem Programm auch ausgenutzt. Während die Floppy schon das Directory liest und die gewünschten Daten in ihren Puffer schreibt, kann der C 64 in aller Ruhe den Bildschirm löschen und neu aufbauen sowie einige andere wichtige Aufgaben erledigen. Eine Schwierigkeit bringt zum Beispiel die Laderoutine mit sich: Der Universal-Lader läuft selbst am Anfang des Basic-Bereichs ab $0801. Nachgeladene Basic-Programme würden also die eigene Lade- und Startroutine überschreiben und sich damit quasi den Ast absägen, auf dem sie selbst sitzen. Um diesen Effekt zu vermeiden, werden die entsprechenden Befehle in die »hinterste Ecke« des $C-Bereiches ($CFE4 bis $CFFF = 53220 bis 53247) kopiert, wo Kollisionen unwahrscheinlich sind. Zu ladende Maschinenprogramme sollten also diesen Bereich nicht überschreiben.

Auch das Kopieren geschieht während die Floppy ihre Daten vorbereitet, ebenso das Initialisieren des RAMs.

An dieser Stelle sei gleich noch eine Besonderheit angesprochen: Der Universal-Lader lädt Programme immer absolut (wie LOAD "Name",8,1), also ab der Speicherstelle, die als Ladeadresse in Byte 2 und 3 des ersten File-Blocks festgelegt ist. Nach dem Laden startet er immer das im


```
Name : universal-lader   0801 0c24
-----------------------------------
0801 : 0b 08 0a 00 9e 32 30 36  3c
0809 : 31 00 00 00 ea ea ea a9  3f
0811 : 01 a2 08 a0 0f 20 ba ff  56
0819 : a9 00 20 bd ff 20 c0 ff  86
0821 : a0 00 84 fb a2 01 20 c9  a8
0829 : ff a9 4d 20 d2 ff a9 2d  82
0831 : 20 d2 ff a9 57 20 d2 ff  b1
0839 : a5 fb 20 d2 ff a9 05 20  e0
0841 : d2 ff a9 22 20 d2 ff a2  9f
0849 : 00 b9 29 0b 20 d2 ff c8  fc
0851 : c0 83 f0 12 e8 e0 22 d0  11
0859 : f0 20 cc ff a5 fb 18 69  fa
0861 : 22 85 fb 4c 25 08 20 cc  7b
0869 : ff a2 01 20 c9 ff a9 55  ec
0871 : 20 d2 ff a9 43 20 d2 ff  b0
0879 : 20 cc ff a9 00 8d 20 d0  c3
0881 : 8d 21 d0 a9 0e 8d 86 02  74
0889 : 20 44 e5 a0 00 b9 ac 0b  ef
0891 : 20 d2 ff c8 c0 29 d0 f5  b8
0899 : 18 a2 18 a0 01 20 0a e5  22
08a1 : a0 00 b9 d5 0b 20 d2 ff  67
08a9 : c8 c0 27 d0 f5 a0 00 a9  6d
08b1 : 07 85 fb a9 0c 85 fc a9  e3
08b9 : e4 85 fd a9 cf 85 fe b1  9d
08c1 : fb 91 fd c8 c0 1d d0 f7  45
08c9 : a9 70 85 fe a9 04 85 fd  b8
08d1 : 20 d9 0a a9 05 85 fc a9  59
08d9 : eb 85 fb a2 01 20 c9 ff  12
08e1 : a2 00 bd fc 0b 20 d2 ff  8f
08e9 : e8 e0 06 d0 f5 20 cc ff  70
08f1 : a2 01 20 c6 ff a2 00 20  4a
08f9 : cf ff 20 c3 0a 9d 0d 04  12
0901 : e8 e0 12 d0 f2 20 cf ff  67
0909 : 20 c3 0a 8d 23 04 20 cf  b2
0911 : ff 20 c3 0a 8d 24 04 a2  a2
0919 : 14 a0 00 20 cf ff c9 ff  a6
0921 : f0 67 91 fd c8 e8 20 cf  dd
0929 : ff 91 fd c8 e8 20 cf ff  58
0931 : 91 fd c8 e8 20 cf ff 91  b4
0939 : fd 20 c3 0a 91 fb e8 c8  a7
0941 : c0 13 d0 f0 a5 fd 18 69  5a
0949 : 13 85 fd 90 02 e6 fe a5  4f
0951 : fb 18 69 28 85 fb 90 02  36
0959 : e6 fc a5 fc c9 07 d0 b9  52
0961 : a5 fb c9 82 90 b3 a0 00  f0
0969 : 20 cf ff c9 ff f0 1a 91  bd
0971 : fd e8 d0 03 20 f0 0a c8  ba
0979 : c0 13 d0 ec a5 fd 18 69  12
0981 : 13 85 fd 90 e1 e6 fe b0  9b
0989 : dd 20 d9 0a 20 cc ff a5  e2
0991 : fd 38 e9 8f 8d 00 70 a5  fc
0999 : fe e9 01 8d 01 70 a9 04  c0
09a1 : 8d 02 70 a9 70 8d 03 70  e1
09a9 : a2 11 ad 02 70 85 fd ad  06
09b1 : 03 70 85 fe a0 be b1 fd  f0
09b9 : c9 4e f0 4d a9 04 85 fc  5a
09c1 : a9 5b 85 fb a0 03 b1 fd  dd
09c9 : 20 c3 0a 91 fb c8 c0 13  af
09d1 : d0 f4 a5 fd 18 69 13 85  68
09d9 : fd 90 02 e6 fe a5 fb 18  b9
09e1 : 69 28 85 fb 90 02 e6 fc  ee
09e9 : a5 fc c9 07 d0 d6 a5 fb  b2
09f1 : c9 82 90 d0 a2 03 a9 21  65
09f9 : 9d eb d9 e8 e0 13 d0 f8  fb
0a01 : 20 cc ff 84 ba 20 e4 ff  58
0a09 : aa e0 91 f0 64 e0 11 f0  19
0a11 : 63 e0 51 f0 54 e0 03 f0  91
0a19 : 50 e0 20 d0 03 4c 10 08  de
0a21 : e0 0d d0 e1 a9 02 a2 08  3d
0a29 : a0 01 20 ba ff ad 02 70  00
0a31 : 18 69 be 85 fd aa 90 03  dc
0a39 : ee 03 70 ad 03 70 85 fe  42
0a41 : 48 a0 01 a9 4c 8d 0d 08  c4
0a49 : b1 fd 8d 0e 08 c8 b1 fd  a7
0a51 : 8d 0f 08 e8 e8 e8 68 a8  4e
0a59 : a9 10 20 bd ff 20 44 e5  a8
0a61 : a9 00 a2 ff a0 ff 4c e4  b8
0a69 : cf a9 01 20 c3 ff 4c 44  47
0a71 : e5 4c 9a 0a ad 03 70 cd  b5
0a79 : 01 70 f0 13 b0 3f ad 02  11
0a81 : 70 18 69 13 8d 02 70 90  86
0a89 : 2e ee 03 70 4c ab 09 ad  9f
0a91 : 02 70 cd 00 70 b0 26 90  85
0a99 : e5 ad 03 70 c9 70 90 1d  c0
0aa1 : d0 07 ad 02 70 c9 17 90  73
0aa9 : 11 ad 02 70 38 e9 13 8d  5a
0ab1 : 02 70 b0 03 ce 03 70 4c  d7
0ab9 : ab 09 4c 06 0a a2 91 4c  51
0ac1 : ab 09 c9 3f b0 01 60 c9  73
0ac9 : 60 b0 03 e9 3f 60 e9 20  5e
0ad1 : c9 80 b0 01 60 e9 60 60  be
0ad9 : a2 00 a0 00 a9 20 20 c3  47
0ae1 : 0a 91 fd e6 fd d0 02 e6  4c
0ae9 : fe e8 e0 be d0 f3 60 98  cb
0af1 : 48 20 cc ff a2 01 20 c9  c3
0af9 : ff a2 00 bd 02 0c 20 d2  a8
0b01 : ff e8 e0 05 d0 f5 20 cc  24
0b09 : ff a2 01 20 c9 ff a2 00  c5
0b11 : bd fc 0b 20 d2 ff e8 e0  a6
0b19 : 06 d0 f5 20 cc ff a2 01  62
0b21 : 20 c6 ff a2 00 68 a8 60  9f
0b29 : 20 42 d0 a9 12 85 06 a2  7e
0b31 : 00 86 07 a9 80 85 00 a5  eb
0b39 : 00 30 fc c9 01 d0 66 bd  75
0b41 : 90 03 9d 00 04 e8 e0 14  ed
0b49 : d0 f5 ad 00 03 f0 56 85  9b
0b51 : 06 ad 01 03 85 07 a9 80  07
0b59 : 85 00 a5 00 30 fc a9 03  df
0b61 : 85 3c a9 02 85 3b a0 00  64
0b69 : b1 3b f0 30 a0 16 b1 3b  f2
0b71 : 30 2a 9d 00 04 e8 c8 b1  2c
0b79 : 3b 9d 00 04 e8 c8 b1 3b  15
0b81 : 9d 00 04 e8 a0 03 b1 3b  9c
0b89 : 9d 00 04 e8 d0 09 8c f0  ae
0b91 : 05 60 a2 00 ac f0 05 c8  67
0b99 : c0 13 d0 ea a5 3b 18 69  dc
0ba1 : 20 90 c1 b0 a5 a9 ff 9d  73
0ba9 : 00 04 60 1e 20 20 44 49  2e
0bb1 : 53 4b 2d 4e 41 4d 45 3a  c7
0bb9 : 20 20 20 20 20 20 20 20  b9
0bc1 : 20 20 20 20 20 20 20 20  c1
0bc9 : 20 20 20 49 44 3a 20 20  01
0bd1 : 20 20 20 20 4d 49 54 20  be
0bd9 : 43 55 52 53 4f 52 54 41  21
0be1 : 53 54 45 4e 20 41 55 53  82
0be9 : 57 41 45 48 4c 45 4e 20  a4
0bf1 : 55 4e 44 20 52 45 54 55  ce
0bf9 : 52 4e 9e 4d 2d 52 00 04  31
0c01 : 00 4d 2d 45 6a 05 20 d5  97
0c09 : ff 86 2d 84 2e 20 33 a5  23
0c11 : a9 01 20 c3 ff 20 44 e5  99
0c19 : 68 68 a9 a7 48 a9 ae 48  32
0c21 : 4c 59 a6 4e 45 4e 53 50  42
```

Listing 6. »UNIVERSAL-LADER« ermöglicht menügesteuertes Laden der Programme

Speicher befindliche Basic-Pogramm. Wurde ein Maschinenprogramm beispielsweise nach $9000 geladen, steht danach immer noch der Universal-Lader am Basic-Start. Damit trotzdem das Maschinenprogramm automatisch gestartet werden kann, installiert der Universal-Lader vor dem Ladevorgang (ab Zeile 4842) einen Sprungbefehl am Anfang des eigenen Maschinenteils, nämlich anstelle der drei NOP-Befehle ab $080D = 2061. Im Programm wird diese Adresse mit $4C überschrieben, dem Code für den Maschinenbefehl JMP. Dann kommen die beiden Bytes für die Startadresse. So erfolgt nach dem Start des Basic-Kopfes (10 SYS 2061) über diesen Befehl der Einsprung in die geladene Routine.

Manchmal ist es sinnvoll, die geladene Routine nicht zu starten, sondern zum Menü zurückzukehren, um vielleicht noch andere Dateien zu laden. In einem solchen Fall sollte man als Startadresse 2064 (= $0810) angeben. Der Universal-Lader startet sich dann selbst noch einmal. Voraussetzung ist natürlich, daß er nicht von der geladenen Routine überschrieben wurde. Bei Basic-Programmen gibt es keine Probleme. Sie überschreiben ohnehin den installierten Zeiger, so daß er wirkungslos bleibt.

```
10 REM  *************************         <144>
20 REM  *                       *         <069>
30 REM  *    UNIVERSAL LADER    *         <216>
40 REM  *                       *         <089>
50 REM  *    INITIALISIERUNG    *         <023>
60 REM  *                       *         <109>
70 REM  *                       *         <119>
80 REM  *    VON UWE GERLACH    *         <205>
85 REM  *                       *         <134>
90 REM  *************************         <224>
95 REM                                    <157>
200 PRINT"{CLR}"CHR$(14): POKE 53280,0: PO
    KE 53281,0                            <225>
210 PRINT"{LIG.BLUE,5SPACE}UNIVERSALLADER
    INITIALISIERUNG"                      <151>
220 PRINT:PRINT"{GREY 2}DIESES PROGRAMM ER
    GAENZT DEN UNIVERSAL-"                <252>
222 PRINT"LADER, DER EIN AUTOMATISCHES AUS
    WAHL-"                                <139>
224 PRINT"MENUE JEDER DISKETTE ERSTELLT." <246>
226 PRINT"ER SOLLTE IMMER AN ERSTER STELLE
     DES"                                 <047>
227 PRINT"DIRECTORYS STEHEN, UM LEICHT AUF
    GERUFEN"                              <124>
228 PRINT"WERDEN ZU KOENNEN."             <219>
229 PRINT"BEIDE PROGRAMME WURDEN GESCHRIEB
    EN VON:"                              <017>
230 PRINT:PRINT,"{GREEN,2SPACE}UWE GERLACH
    ",,,"{2SPACE}BRUEHLSTR. 23",,,"{2SPACE
    }6440 BEBRA 1{GREY 2}"                <008>
240 PRINT:PRINT"DIESER TEIL DIENT ZUM EINM
    ALIGEN FESTLE-";                      <195>
250 PRINT"GEN DER EINTRAEGE, DIE BEIM STAR
    TEN DES"                              <192>
260 PRINT"UNIVERSALLADERS AUFGEFUEHRT WERD
    EN SOL-"                              <166>
270 PRINT"LEN. (MIT{SPACE,WHITE}J{GREY 2},
    {SPACE,WHITE}N{GREY 2,SPACE}UND{SPACE,
    WHITE}S{GREY 2,SPACE}AUSWAEHLEN."     <013>
280 PRINT"{WHITE}<RET.>{GREY 2,SPACE}FUER
    ALTEN ZUSTAND,{SPACE,WHITE}↑{GREY 2,SP
    ACE}GEHT ZU-"                         <248>
290 PRINT"RUECK UND{SPACE,WHITE}Q{GREY 2,S
    PACE}BRICHT AB.)"                     <049>
300 DIM N(144)                            <178>
310 POKE 649,1: REM  TASTATURPUFFER       <203>
320 POKE 650,128: REM  TASTATURPUFFER     <218>
400 PRINT:PRINT"{LIG.BLUE}BITTE DISK EINLE
    GEN UND TASTE DRUECKEN !{GREY 2}"     <096>
500 GET A$: IF A$="" THEN 500             <247>
510 PRINT: PRINT CHR$(142);"{CLR,GREEN}TIT
    EL DIESER DISKETTE:{SPACE,RVSON}";: RE
    M  REVERSE ON                         <088>
1000 OPEN 1,8,15     : REM  KOMMANDOS     <223>
1010 OPEN 2,8,2,"#" : REM  PUFFER         <204>
1020 PRINT#1, "U1 2 0 18 0"               <062>
1030 PRINT#1, "B-P 2 144"                 <197>
1040 FOR J=1 TO 16                        <184>
1050 : GET#2, A$                          <094>
1060 : IF A$=CHR$(160) THEN 2000          <052>
1070 : PRINT A$;                          <080>
1080 NEXT J                               <156>
2000 PRINT:PRINT"{RVOFF,GREY 2}": REM  REV
     ERSE OFF                             <144>
2020 T=18            : REM  TRACK         <018>
2030 S=1             : REM  SECTOR        <127>
3000 PRINT#1, "U1: 2 0";T;S: REM  LESEN   <243>
3010 FOR J=0 TO 224 STEP 32               <159>
3011 : PRINT#1, "B-P:"2,STR$(J+2)         <041>
3014 : GET#2, A$: REM  FILETYP HOLEN      <039>
3016 : IF A$="" THEN 3690                 <254>
3020 : PRINT#1, "B-P:"2,STR$(J+24)        <028>
3030 : GET#2, A$: REM  MARKIERUNG HOLEN   <031>
3035 : M$=" JA ": V$="J"                  <014>
3040 : IF A$=CHR$(128) THEN M$="NEIN": V$=
     A$                                   <110>
3042 : IF A$="N" THEN M$="SKIP": V$="N"   <056>
3045 : M$="{BLUE}"+M$+"{4LEFT,RVOFF,GREY 2
     }"                                   <244>
3050 : PRINT#1, "B-P:"2,STR$(J+5)         <092>
3060 : N$=""                              <149>
3070 : FOR K=1 TO 16                      <077>
3080 :  GET#2, A$: REM  FILENAMEN HOLEN   <167>
3090 :  IF A$=CHR$(160) THEN 3300         <243>
3100 :  N$=N$+A$                          <135>
3110 : NEXT K                             <238>
3300 : PRINT "{LIG.BLUE}";N$; TAB(18); "{W
     HITE,RVSON}(J/N/S){RVOFF,SPACE}";    <005>
3330 : GET A$                             <169>
3332 : IF J<>0 THEN IF A$="↑" THEN J=J-32:
      PRINT"{4SPACE,2UP}"  : GOTO 3011    <132>
3335 : IF A$="Q" THEN 5000                <171>
3337 : Z=Z+1: IF Z>6 THEN PRINT"{RVSON}"; <236>
3339 : PRINT M$;: IF Z>12 THEN Z=0        <224>
3345 : IF A$="J" OR A$="Y" THEN 7050      <017>
3347 : IF A$="" THEN GOTO 3330            <155>
3350 : IF A$="S" THEN V$="N": PRINT"SKIP":
      GOTO 3400: REM UEBERSPRINGEN        <081>
3360 : IF A$=CHR$(13) THEN PRINT M$: GOTO
     3400: BEI RETURN                     <088>
3370 : V$=CHR$(128): PRINT"NEIN"          <223>
3400 : PRINT#1, "B-P:"2,STR$(J+24)        <156>
3410 : PRINT#2, V$;: REM  MARKIEREN       <002>
3690 NEXT J                               <226>
3700 PRINT#1, "U2 2 0" T;S                <200>
4000 PRINT#1, "B-P: 2 0"                  <004>
4010 GET#2, A$, B$                        <177>
4020 T=ASC(A$+CHR$(0))                    <156>
4030 S=ASC(B$+CHR$(0))                    <226>
4040 IF T<>0 THEN 3000: REM  NEXT BLOCK   <060>
5000 CLOSE 1: CLOSE 2                     <252>
6000 PRINT:PRINT:PRINT"NOCH EINE DISKETTE?
     "                                    <174>
6010 GET A$: IF A$="" THEN 6010           <121>
6020 IF A$="J" THEN PRINT"{CLR}";CHR$(14):
      GOTO 400                            <017>
6050 END                                  <210>
7000 :                                    <118>
7010 REM ** STARTADRESSE FESTSTELLEN **   <139>
7020 :                                    <138>
7050 PRINT" JA ";                         <040>
7070 OPEN 3,8,3,N$: REM FILE ZUM LESEN OEF
     FNEN                                 <043>
7100 GET#3,SL$: SL=ASC(SL$+CHR$(0)): REM S
     TARTBYTE LOW                         <084>
7110 GET#3,SH$: SH=ASC(SH$+CHR$(0)): REM S
     TARTBYTE HIGH                        <200>
7190 CLOSE 3                              <105>
7210 PRINT"SA "; LEFT$(STR$(SH*256+SL)+"{5
     SPACE}",6);                          <032>
7220 INPUT"{7LEFT}";SA: REM STARTADRESSE  <005>
7250 SH=INT(SA/256): REM HIGHBYTE         <214>
7260 SL=SA-SH*256: REM LOWBYTE            <090>
7900 V$=CHR$(0)+CHR$(SL)+CHR$(SH): REM MAR
     KIERUNG VORBEREITEN                  <204>
7990 GOTO 3400                            <136>
```

Listing 7. »UNIV.LADER.INIT« präpariert die Disketten für menügesteuertes Laden

Der Universal-Lader bereitet dann das Lesen des Floppy-Puffers vor. Der M-R-Befehl wird automatisch erst ausgeführt, nachdem die Floppy fertig ist. Nun werden zunächst der Disk-Name und die ID geholt und angezeigt. Die Unterroutine WANDEL besorgt die Umwandlung von ASCII- in Bildschirm-Code. Danach erfolgt das Einlesen der Filenamen ins RAM. Soviel wie auf den Bildschirm passen, werden sofort angezeigt, damit auch während des Einlesens sehr langer Verzeichnisse schon etwas zu sehen ist. Sofort nach der Übernahme des letzten Namens erscheint die erwähnte farbige Markierung und das Programm wartet auf Tastatureingaben.

Außer den Tasten <CUSOR aufwärts/abwärts> zum Scrollen beziehungsweise <RETURN> zum Auswählen, sind noch zwei weitere Funktionen implementiert: <SPACE> erzeugt ein neues Menü und ermöglicht damit den unproblematischen Diskettenwechsel; <Q> oder <RUN/STOP> brechen das Programm ab.

Die Initialisierung

Noch ein paar Worte zum Initialisierungsprogramm UNIV.LADER.INIT (Listing 7). Es ist in Basic geschrieben und dient zum einmaligen Festlegen des File-Status im Menüprogramm. Selbstverständlich braucht es nicht auf jeder Diskette gespeichert zu sein. Bei unbearbeiteten Disketten werden alle Filenamen angezeigt.

Mit dem Initialisierungsprogramm lassen sich die Markierungen sowohl anbringen, als auch wieder entfernen. Dazu wird jeder Eintrag und sein Status einzeln angezeigt. Bei Angabe von <J> (Ja) oder <Y> (Yes) erscheint er im Menü des Universal-Laders und ist auch anwählbar, bei <N> (Nein) erscheint er nicht und bei <S> (Skip) erscheint er zwar, wird aber vom Cursor übersprungen. <RETURN> übernimmt den jeweils voreingestellten Status ohne Änderungen.

Bei <J> wird auch nach der Einsprungadresse gefragt. Sie ist nur bei Maschinenprogrammen relevant und muß dezimal eingegeben werden. Zur Erleichterung der Bedienung wird jeweils die Ladeadresse angezeigt. Sie ist meistens mit der Einsprungadresse identisch und kann einfach mit <RETURN> übernommen werden.

Jeweils nach acht Einträgen werden die Änderungen gespeichert. Drücken von <Q> bricht das Programm ab.

Service bitte!

Den reinen Austausch von Informationen zwischen C64 und Floppystation haben Sie jetzt im Griff. Daß der C64 aber bei einem Bus-Zugriff auf die Floppy angehalten wird, solange dort noch ein Interrupt-Programm abläuft, bedeutet gleichzeitig, daß wertvolle Rechenzeit veschenkt wird. Wie könnte nun ein möglichst unabhängiges Programmierkonzept aussehen, bei dem der C64 nicht ständig in Warteschleifen hängt?

Die Antwort scheint einfach: Am seriellen Bus gibt es eine Leitung mit der Bezeichnung »SRQ«. Dieses Kürzel steht für »Service Request« (Bedienungsanforderung). Jedes Gerät am Bus kann theoretisch diesen Anschluß auf LOW ziehen, um dem C64 mitzuteilen, daß es abgefragt werden möchte. SRQ wirkt im C64 (genau wie das RD-Signal des Kassetten-Ports) auf den FLAG-Eingang des CIA 1 und kann zur Erzeugung eines maskierbaren Interrupts (IRQ) dienen. So gut ausgedacht dieses Konzept aber ist, wird es von keinem der üblichen C64-Peripheriegeräte am seriellen Bus benutzt. Selbst die Floppy VC1541 hat keinerlei Möglichkeit, die SRQ-Leitung zu beeinflussen, da

```
10 rem open 1,8,1,"0:floppy srq.obj"
12 rem open 2,4
13 sys 32768; einsprung fuer neuen profiass
14 ;
15 ; **************************
16 ; * floppy service request *
17 ; *                        *
18 ; *    von  uwe gerlach    *
19 ; *   tel.  06151/316205   *
20 ; *    bzw.  06622/3908    *
21 ; *                        *
22 ; *        13.06.86        *
23 ; **************************
24 ;
25 ;
26 ; ueber data-line des seriellen bus
27 ; an den computer
30 ;
31 ;
100 .tit "service request c64 & vc1541 "
120 .opt oo
125 ;
130 *=$0500; im userpuffer der floppy
140 ;
500 start lda $1c00; port b des dc abfragen
510 and #%00010000; lichtschrankenbit isolieren
520 beq aus
530 :
600 ein lda $1c00;port b des dc abfragen
610 ora #%00001000; led-bit setzen
620 sta $1c00; led einschalten
650 ;
660 lda $1800; port b des bc abfragen
670 and #%11111101; data out-bit loeschen
680 sta $1800; data out nach low ziehen
690 jmp start
695 ;
700 aus lda $1c00;port b des dc abfragen
710 and #%11110111; led-bit loeschen
720 sta $1c00; led ausschalten
750 ;
760 lda $1800; port b des bc abfragen
770 ora #%00000010; data out-bit setzen
780 sta $1800; data out freigeben
790 jmp start
```

Listing 8. »FLOPPY SQR« erzeugt Interrupt-Anforderungen von Seiten der Floppy

diese gar nicht erst an die Elektronik angeschlossen wurde, sondern lediglich zur zweiten Buchse durchgeschleift ist.

Mit der vorhandenen Bus-Hardware ist aber ein ähnlicher Einsatz realisierbar, wenn auf die Interrupt-Möglichkeit verzichtet wird. Das Beispielprogramm FLOPPY SRQ (Listing 8) mißbraucht die Datenleitung des seriellen Busses als Meldeleitung, indem sie von der Floppy zur Signalisierung eines Unterbrechungswunsches auf 0 gezogen wird. Der C64 kann nun beispielsweise während seiner normalen Interruptroutine zyklisch nachsehen, welchen Status die Datenleitung hat. 1 ist der Normalzustand. Erkennt der Rechner jedoch eine 0, spricht er die Floppy an, worauf diese den Bus wieder freigeben muß, und holt sich die bereitgestellten Daten ab.

Das genannte Beispielprogramm läuft in der Floppy und überträgt zu Demonstrationszwecken lediglich den Zustand der Schreibschutz-Lichtschranke auf die Datenleitung. Außerdem wird zur Kontrolle noch die rote Busy-LED der Floppy angesteuert. Ist die Lichtschranke offen, dann leuchtet die LED und die Datenleitung befindet sich auf 0. Genau umgekehrt liegen die Verhältnisse bei unterbrochener Lichtschranke.

Mit dieser Abfragemethode lassen sich effiziente Programme schreiben, die wirkungsvoll die Möglichkeiten des Multiprozessings ausschöpfen. Es ist empfehlenswert, zumindest den gesamten Datenempfang innerhalb der Interruptschleife zu plazieren. (Uwe Gerlach/sk)

Literaturhinweise:
Karsten Schramm. Die Floppy 1541, Markt & Technik Verlag AG, ISBN Nr 3-89090-098-4, 49 Mark

Vom Fuhrwerk zum GTI: die getunte Floppy 1541

Kein Thema bewegt die Gemüter der 64'er-Leser so sehr wie die Geschwindigkeit der Floppy 1541. Mit gutem Grund befaßten sich mehrere Listings des Monats mit diesem Thema wie Hypra-Load, EXOS oder Copy +. Schauen wir einmal hinter die Kulissen dieser schnellen »Bit-Schieber«.

Die normale Arbeitsgeschwindigkeit der Floppy-Station 1541 ist Grund genug, das Prinzip der schnellen Datenübertragung einmal genau unter die Lupe zu nehmen. Ich möchte Ihnen ein paar schnelle und universell einsetzbare Assembler-Routinen präsentieren, die die Übertragung der Daten bei kombiniertem Einsatz bis zu 15mal beschleunigen können.

Zunächst muß man sich verdeutlichen, warum die 1541 so langsam ist. Allgemein wird die serielle Übertragung, das bitweise Senden eines Bytes über eine Datenleitung, als Hauptursache dargestellt. Tatsächlich aber sind umständlich programmierte DOS-Routinen der wahre Grund für die langsame Übertragung. Auch die laufwerksinternen Routinen kann man deutlich beschleunigen, um den Weg eines Bytes vom Schreib-/Lesekopf zum seriellen Bus zu beschleunigen.

Der serielle Bus

Zunächst wollen wir uns jedoch mit dem seriellen Bus beschäftigen. Um die Wirkung der Fast-Routinen zu verstehen, wollen wir einmal sehen, wie eine »normale« Übertragung funktioniert.

Bei jeder Übertragung gibt es ein Gerät, das die Daten sendet (»Talker«), sowie eines oder mehrere, das (die) die Daten empfängt (empfangen). Ein solches Gerät wird »Listener« genannt. Während ein Drucker nur Listener sein kann, können Floppy und Computer beide Betriebsarten ausüben. Das Verbindungskabel vom Computer zu den Peripheriegeräten besitzt sechs Leitungen, von denen drei für die serielle Übertragung wichtig sind. Diese Leitungen kennen die logischen Zustände »0« und »1«. Leitung 3 wird »Attention« (ATN) genannt und wird »1« (»high«), wenn der Computer einen Befehl übermitteln will. Alle angeschlossenen Geräte erwarten dann die Geräteadresse, um feststellen zu können, ob sie oder ein anderes Gerät adressiert werden sollen. Leitung 4 heißt »Clock« und dient dazu, die Leitung 5 (»DATA«) zu takten. Letztere stellt die Datenleitung dar, über welche die acht Bit eines Bytes nacheinander übertragen werden.

An dieser Stelle taucht die Frage auf, warum man jedem Datenbit einen Impuls der Clock-Leitung mit auf den Weg gibt. Würde man auf diese Taktung verzichten, müßten Listener und Talker eine »Vereinbarung« über die zeitlichen Abstände der Datenbits treffen, die auf einen Taktzyklus genau abgestimmt sein müßte, um eine falsche Übermittlung zu verhindern. Dies ist jedoch nicht möglich, da der Prozessor des C 64 jederzeit unkontrollierbar vom Video-Interface-Chip (VIC) unterbrochen werden kann, um dann für 40 Taktzyklen angehalten zu werden. In die-

a) vom Computer zur Floppy							
Computerroutine				Floppyroutine			
	LDA	11			LDX	16	
	STA	$DD00	;ATN setzen	L1	LDA	$1800	;auf ATN
	LDX	03			BPL	L1	;warten
L1	LDA	$DD00	;auf DATA		STX	$1800	;DATA setzen
	BPL	L1	;warten		NOP		;auf Computer
	STX	$DD00	;ATN rücksetzen		NOP		;warten
L2	LDA	Puffer,Y	;Datenbyte holen		LDX	00	
	LDX	11			STX	$1800	;DATA rücksetzen
	STX	$DD00	;ATN setzen		NOP		;auf Computer
	TAX		;Datenbyte sichern		NOP		;warten
	AND	240	;Bits 4 und 5	L2	LDA	$1800	;auf ATN
	STA	$DD00	;übertragen		BPL	L2	;warten
	LSR	A			NOP		
	LSR	A			NOP		
	AND	240	;Bits 6 und 7		LDX	$1800	;Bits 4 und 5
	STA	$DD00	;übertragen		LDA	TAB45,X	;holen
	TXA		;Datenbyte holen		NOP		
	AND	15			LDX	$1800	;Bits 6 und 7
	TAX				ORA	TAB67,X	;einblenden
	LDA	TAB,X	;Bits 0 und 1		NOP		
	STA	$DD00	;übertragen		NOP		;auf Computer
	LSR	A			NOP		;warten
	LSR	A			LDX	$1800	;Bits 0 und 1
	AND	240	;Bits 2 und 3		ORA	TAB01,X	;einblenden
	STA	$DD00	;übertragen		NOP		
	NOP				LDX	$1800	;Bits 2 und 3
	NOP		;auf Floppy		ORA	TAB23,X	;einblenden
	NOP		;warten		STA	Puffer,Y	;Datenbyte sichern
	INY				INY		
	BNE	L2	;nächstes Byte		BNE	L2	;nächstes Byte
	LDA	03			LDX	15	
	STA	$DD00	;ATN rücksetzen		STX	$1800	;ATN rücksetzen
	RTS				RTS		
TAB	.BYTE	$00,$10,$20,$30		TAB 45	.BYTE	$00,$20	
	.BYTE	$40,$50,$60,$70		TAB 67	.BYTE	$00,$80,$10,$30,$40,$C0	
	.BYTE	$80,$90,$A0,$B0		TAB 01	.BYTE	$00,$02	
	.BYTE	$C0,$D0,$E0,$F0		TAB 23	.BYTE	$00,$08,$01,$03,$04,$0C	

Tabelle 1. Fast-Routinen zur seriellen Übertragung. Der Datenpuffer sowie die Tabellen dürfen nicht Page-übergreifend im Speicher liegen, da sonst bei X/Y-Indizierung ein Taktzyklus mehr benötigt wird. Tabelle 1 wird auf der nächsten Seite fortgesetzt.

ser Zeit holt sich der VIC die Daten für den Bildschirmaufbau aus dem RAM.

Wie gesagt, wird zunächst bei aktivierter ATN-Leitung, die immer ein Kommandobyte ankündigt, die Geräteadresse übertragen. In diesem Byte wird neben der eigentlichen Adresse (Bits 0 bis 4) auch die Funktion des adressierten Gerätes festgelegt:

Bit 5=1, Bit 6=0:	das Gerät wird zum Listener.
Bit 5=0, Bit 6=1:	das Gerät wird zum Talker.
Bit 7=0:	Kennzeichen für Geräteadressen-Byte.

Durch Übermittlung der Geräteadresse 31 wird der Datentransfer eingestellt. Die Bits 5 bis 7 entsprechen der Funktion beim Aufruf des Gerätes, nur wird jetzt eben der jeweilige Zustand beendet:

Bit 5=1, Bit 6=0:	der Betriebszustand Listen wird beendet (»UNLISTEN«).
Bit 5=0, Bit 6=1:	der Betriebszustand Talk wird beendet (»UNTALK«).
Bit 7=0:	siehe oben

Nach der Geräteadresse wird ebenfalls bei gesetzter ATN-Leitung in einem zweiten Kommando-Byte die Sekundäradresse übertragen. Hier steht die Adresse in den Bits 0 bis 3, während die Bits 5 bis 7 immer eins sind. Das vierte Bit zeigt an, ob der Kanal geöffnet (=1) oder geschlossen wird (=0). Als nächstes wollen wir uns schrittweise die Übertragung eines Kommandobytes ansehen:

Zunächst wird ATN High, zum Zeichen, daß ein Befehl folgt. Daraufhin wird die DATA-Leitung auf »0« gesetzt, die Clock-Leitung auf »1«. Die Floppy-Elektronik setzt nun automatisch DATA auf »1«. Geschieht dies nicht innerhalb einer Millisekunde, muß der Computer davon ausgehen, daß kein Gerät angeschlossen ist und gibt die Fehlermeldung »DEVICE NOT PRESENT« aus. Durch die Zeitspanne von einer Millisekunde haben auch die Geräte, die keine automatische Schaltung zum Hochsetzen der DATA-Leitung besitzen, Gelegenheit, auf den Aufruf zu reagieren. Ist das Laufwerk bereit Daten zu empfangen, setzt es DATA wieder auf »0«. Der Pegel dieser Leitung wird genau dann wieder »0«, wenn alle angeschlossenen Geräte diesen Schritt vollzogen haben. Jetzt setzt der Computer die Clock-Leitung auf »0«, um mitzuteilen, daß der Datentransfer beginnen wird. Mit jedem Low-Impuls auf der Clock-Leitung wird ein Bit auf der DATA-Leitung übertragen. Zwischen je zwei Datenbit wird die Datenleitung auf »0« und Clock auf »1« gesetzt. Die angeschlossenen Geräte teilen dem Computer mit, daß sie das Datenbit empfangen haben, indem sie die Datenleitung auf »1« setzten. Falls diese Quittierung nicht erfolgt, wird die Fehlermeldung »Time-out« im Status-Byte des C 64 ($90) vermerkt. Die Übertragung eines Datenbytes erfolgt analog, die ATN-Leitung ist jedoch nicht gesetzt.

Wenn hier von Low, High, 1 und 0 gesprochen wird, so sind immer die logischen Zustände der Leitungen gemeint. In Wirklichkeit sind die Leitungen jedoch im genau umgekehrten Zustand. Der Grund hierfür liegt in der Elektrotechnik, ist jedoch für die Programmierung nicht interessant, da durch vorgeschaltete Inverter am Ein- und Ausgang der Geräte dieser Zustand automatisch umgedreht wird. Es gibt jedoch eine Ausnahme: Der Eingang des C-64 besitzt keinen Inverter. Wie wir noch sehen werden, müssen in diesem Fall alle ankommenden Zustände »per Hand« invertiert werden.

Falls Sie die Übertragung einmal genau studieren möchten, hier folgen die Adressen der einzelnen Routinen:

	C 64	Floppy 1541
Byte auf Bus geben	$ED40-$EDAC	$E909-$E999
Byte vom Bus holen	$EE13-$EE84	$E9C9-$EA2D

b) Von der Floppy zum Computer

Computerroutine

```
      LDX   11
      STX   $DD00      ;ATN setzen
      LDX   03
L1    LDA   $DD00      ;auf DATA
      BPL   L1         ;warten
      STX   $DD00      ;ATN rücksetzen
L2    NOP
      NOP
      LDA   11
      STA   $DD00      ;ATN setzen
      NOP              ;warten auf
      NOP              ;Floppy
      LDX   03
      STX   $DD00      ;ATN rücksetzen
      LDX   $DD00      ;Bits 5 und 7
      LDA   TAB57,X    ;holen
      LDX   $DD00      ;Bits 4 und 6
      ORA   TAB46,X    ;einblenden
      NOP
      LDX   $DD00      ;Bits 1 und 3
      ORA   TAB13,X    ;einblenden
      LDX   $DD00      ;Bits 0 und 2
      ORA   TAB02,X    ;einblenden
      STA   Puffer,Y   ;Datenbyte sichern
      INY
      BNE   L2         ;nächstes Byte
      RTS
```

Floppyroutine

```
      LDX   16
L1    LDA   $1800      ;warten
      BPL   L1         ;auf ATN
      STX   $1800      ;DATA setzen
L2    LDA   Puffer,Y   ;Datenbyte holen
      STA   $C1        ;sichern
      LSR   A
      LSR   A
      LSR   A
      LSR   A
L3    LDX   $1800      ;auf ATN
      BPL   L3         ;warten
      STA   $1800      ;Bits 5 und 7
      ASL   A          ;übertragen
      AND   15         ;Bits 4 und 6
      STA   $1800      ;übertragen
      LDA   $00C1      ;Datenbyte holen
      AND   15         ;Bits 1 und 3
      STA   $1800      ;übertragen
      ASL   A
      AND   15         ;Bits 0 und 2
      STA   $1800      ;übertragen
      INY
      BNE   L2         ;nächstes Byte
      LDX   15
      STX   $1800      ;ATN rücksetzen
      RTS
```

```
TAB02   .BYTE 0,0,0,$05,0,0,0,0
TAB13   .BYTE 0,0,0,$0A,0,0,0,0
TAB46   .BYTE 0,0,0,$50,0,0,0,0
TAB57   .BYTE 0,0,0,$A0,0,0,0,0
        .BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE 0,0,0,$01,0,0,0,0
        .BYTE 0,0,0,$02,0,0,0,0
        .BYTE 0,0,0,$10,0,0,0,0
        .BYTE 0,0,0,$20,0,0,0,0
        .BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE 0,0,0,$04,0,0,0,0
        .BYTE 0,0,0,$08,0,0,0,0
        .BYTE 0,0,0,$40,0,0,0,0
        .BYTE 0,0,0,$80,0,0,0,0
        .BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        .BYTE 0,0,0,$00,0,0,0,0
        .BYTE 0,0,0,$00,0,0,0,0
        .BYTE 0,0,0,$00,0,0,0,0
        .BYTE 0,0,0,$00,0,0,0,0
```

Alle 0-Bytes sind Füllbytes und können mit Informationen beschrieben werden.

Tabelle 1. »Serielle Fastroutinen« (Schluß)

Jetzt wird's schneller

Sie sehen, wie kompliziert eine serielle Übertragung normalerweise aufgebaut ist. Das eigentliche Problem liegt nicht in der bitweisen Übermittlung, sondern in dem langwierigen Handshake-Verfahren. Falls wir dieses kürzen könnten, würde die Übertragung wesentlich beschleunigt. Die Lösung liegt darin, den Bildschirm durch Löschen des vierten Bits im Register 17 des VIC abzuschalten, um damit die Unterbrechungen des Prozessors zu verhindern. Anschließend müssen wir noch etwas warten, um den Bildschirmabbau abzuwarten. Dies bewirkt folgende Routine:

```
    LDA  $D011   ;Adresse Reg. 17
    AND  #$EF    ;Bit 4 löschen
    STA  $DO11   ;Bildschirm aus
    LDY  #$04    ;
L1  DEY          ;warten
    BNE  L1      ;
```

Jetzt ist es möglich, die Routinen in Computer und Floppy auf einen Taktzyklus genau abzustimmen. Zusätzlich muß man noch verlangen, daß keine anderen Geräte am seriellen Bus angeschlossen sind, da diese unsere Übertragung nicht »verstehen« und stören würden. Neben dem Entfallen des Handshake ist die Clock-Leitung überflüssig geworden. Da diese ohne Aufgabe brachliegt, werden wir sie als zweite Datenleitung benutzen und sind damit in der Lage, jeweils 2 Bit parallel zu übermitteln. Da eine Adressierung mit Geräte- und Sekundäradresse ebenfalls nicht mehr erforderlich ist (bei nur einem Gerät am Bus), können wir diese ebenfalls streichen. Das Ergebnis (Tabelle 1) sind also drastisch gekürzte Routinen, die zirka siebenmal schneller übertragen als die Originalroutinen. Allein durch die Parallel-Übertragung von 2 Bit hätte man die Geschwindigkeit nur verdoppeln können, was verdeutlicht, wie sehr die Übertragung durch die Software gebremst wird. Um die vorgestellten Routinen zu verstehen, muß man wissen, wie man die Zustände der Leitungen softwaremäßig beeinflussen kann. Hierzu existiert im Register 0 der CIA 1 der Floppy (ab $1800), beziehungsweise im gleichen Register der CIA 2 im C 64 (ab $DD00), für jede Leitung jeweils 1 Bit für den Eingangs- und Ausgangszustand:

	C 64	Floppy 1541
Bit 0:	1	DATA (Eingang)
Bit 1:	1	DATA (Ausgang)
Bit 2:	0	CLOCK (Eingang)
Bit 3:	ATN (Ausgang)	CLOCK (Ausgang)
Bit 4:	CLOCK (Ausgang)	ATN (Ausgang)
Bit 5:	DATA (Ausgang)	Geräteadresse
Bit 6:	CLOCK (Eingang)	Geräteadresse
Bit 7:	DATA (Eingang)	ATN (Eingang)

Durch Setzen beziehungsweise Löschen der entsprechenden Bits erhält man die gewünschten Zustände. Beim Auslesen muß man, wie bereits erwähnt wurde, beim C 64 das gelesene Eingangsbit invertieren.

Bevor Sie die schnellen Routinen nutzen können, müssen Sie den Bildschirm abschalten und natürlich auch den Systeminterrupt mit dem SEI-Befehl sperren. Die NOP-Befehle dienen dazu, einen gleichen zeitlichen Verlauf in Computer und Floppy zu erhalten. Sie können mit diesen Routinen 1 bis 256 Byte übertragen, wobei das Y-Register mit dem reziproken Wert geladen werden muß:

Für die Übertragung eines Byte müssen Sie das Register mit einer 255 laden, bei 256 Byte mit einer Null. Vor dem Aufruf muß weiterhin die Startadresse des Puffers gesetzt werden, aus dem Sie die Daten übertragen beziehungsweise in den Sie die geladenen Bytes speichern wollen. Da die Funktion der Programme nicht unmittelbar einsichtig ist, habe ich diese in Tabelle 2 schrittweise anhand eines Beispiels dokumentiert.

Zu einer weiteren Steige-

Computerroutine

Befehl		Akku	X-Register
LDA	203	11001011	-
TAX		11001011	11001011
AND	240	11000000	11001011
STA	$DD00	11000000	11001011
LSR	A	01100000	11001011
LSR	A	00110000	11001011
AND	240	00110000	11001011
STA	$DD00	00110000	11001011
TXA		11001011	11001011
AND	15	00001011	11001011
TAX		00001011	00001011
LDA	TAB,X	10110000	00001011
STA	$DD00	10110000	00001011
LSR	A	01011000	00001011
LSR	A	00101100	00001011
AND	240	00100000	00001011
STA	$DD00	00100000	00001011

Floppyroutine

Befehl		Akku	X-Register
LDX	$1800	-	00000000
LDA	TAB45,X	00000000	00000000
NOP		00000000	00000000
LDX	$1800	00000000	00000101
ORA	TAB67,X	11000000	00000101
NOP		11000000	00000101
NOP		11000000	00000101
NOP		11000000	00000101
LDX	$1800	11000000	00000101
ORA	TAB01,X	11000011	00000101
NOP		11000011	00000101
LDX	$1800	11000011	00000001
ORA	TAB23,X	11001011	00000001

Computerroutine

Befehl		Akku	Zelle $C1
LDA	217	11011001	—
STA	$C1	11011001	11011001
LSR	A	01101100	11011001
LSR	A	00110110	11011001
LSR	A	00011011	11011001
LSR	A	00001101	11011001
STA	$1800	00001101	11011001
ASL	A	00011010	11011001
AND	15	00001010	11011001
STA	$1800	00001010	11011001
LDA	$00C1	11011001	11011001
AND	15	00001001	11011001
STA	$1800	00001001	11011001
ASL	A	00010010	11011001
AND	15	00000010	11011001
STA	$1800	00000010	11011001

Computerroutine

Befehl		Akku	X-Register
LDX	$DD00	—	10000000
LDA	TAB57,X	10000000	10000000
LDX	$DD00	10000000	00000000
ORA	TAB46,X	11010000	00000000
NOP		11010000	00000000
LDX	$1800	11010000	10000000
ORA	TAB13,X	11011000	10000000
LDX	$DD00	11011000	01000000
ORA	TAB02,X	11011001	01000000

Die unterstrichenen Bits werden übertragen.

Tabelle 2. Hier ist die Übertragung eines Bytes (203) vom Computer zur Floppy sowie eines Bytes (217) von der Floppy zum Computer dargestellt

rung der Übertragungsgeschwindigkeit ist der serielle Bus nicht mehr geeignet. Man muß daher auf eine andere Möglichkeit zurückgreifen: Wenn man sich die CIAs, die den seriellen Bus steuern, ansieht, erkennt man, daß sich in den jeweiligen Registern 1 (Floppy: $1801, C-64: $DD01) ein unbenutzter Ein-/Ausgabeport mit acht frei programmierbaren Datenleitungen befindet. Wenn man diese beiden Ports mit einem Kabel verbinden würde, könnte man so nicht nur 1 oder 2, sondern 8 Bit, also ein ganzes Byte, gleichzeitig übertragen. Diese Verbindung wird allgemein als »Parallelbus« bezeichnet und von fast allen professionellen Floppy-Speedern benutzt. Da die Leitungen der CIA 2 des C 64 am User-Port herausgeführt sind, braucht man nur in der Floppy einen Adaptersockel einzusetzen. Durch die Datenrichtungsregister ($1803 und $DD03) kann man jede einzelne Leitung wahlweise auf Ein- oder Ausgang programmieren. Da wir ganze Bytes übertragen wollen, werden natürlich alle acht Leitungen eines Ports gleichzeitig je nach Funktion auf Ein- oder Ausgabe geschaltet.

Man sieht sofort, daß die nötigen Routinen ungleich einfacher sind, als die zur Steuerung des seriellen Busses erforderlichen:

Vom C 64 zur 1541	Von der 1541 zum C 64
Im C 64:	**Im C 64:**
LDA #$FF ;Richtungsreg.	LDA #$00 ;Richtungsreg.
STA $DD03 ;auf Ausgang	STA $DD03 ;auf Eingang
LDA WERT ;Byte	LDA $DD01 ;Byte holen
STA $DD01 ;ausgeben	STA WERT ;
In der 1541:	**In der 1541**
LDA #$00 ;Richtungsreg.	LDA #$FF ;Richtungsreg.
STA $1803 ;auf Eingang	STA $1803 ;auf Ausgang
LDA $1801 ;Byte holen	LDA WERT ;Byte
STA WERT ;	STA $1801 ;ausgeben

Das einzige Problem der Übertragung liegt hier ebenfalls in der zeitlichen Synchronisation. Wie bei den Fast-Routinen des seriellen Busses wollen wir auch hier dazu die ATN- und DATA-Leitung benutzen. Theoretisch könnte man übrigens auch die Clock-Leitung verwenden. Die beiden anderen Leitungen stellen jedoch am Eingang der Geräte jeweils die siebten Bits dar und lassen sich daher software-seitig komfortabel mit den Befehlen BPL und BMI abfragen. Dies ist der einzige Grund, der gegen die Clock-Leitung spricht. In Tabelle 3 sehen Sie die »gebrauchsfertigen« Parallelroutinen, die extrem kurz und schnell sind. So konnte die Geschwindigkeit im Vergleich zu den Fast-Routinen aus Tabelle 1 nochmals um den Faktor 5 gesteigert werden. Da der serielle Bus benutzt wird, müssen auch hier vor dem Start Systeminterrupt und Bildschirm abgeschaltet werden. Das Y-Register bestimmt wiederum die Zahl der zu übertragenen Bytes.

Um ein Disketten-Byte zu lesen, werden je nach Aufzeichnungsrate zwischen 26 und 32 Mikrosekunden benötigt. In jedem Fall ist die Zeit zwischen 2 Byte aber viel zu kurz, um 1 Byte seriell zu übertragen, woran auch die Fast-Routinen nichts ändern können. Die Floppy ist daher mit einem RAM-Speicher ausgerüstet, in dem die Bytes zwischengespeichert und anschließend geschlossen übertragen werden. Durch die Parallelübertragung läßt sich hingegen ein gelesenes Byte übertragen, noch bevor das nächste am Schreib-/Lesekopf der Floppy anliegt. Damit kann man einen Track während einer Umdrehung lesen. Theoretisch kann man daher bei einer Laufwerksgeschwindigkeit von 300 U/min einen Track in einer Fünftelsekunde lesen oder beschreiben. Floppy-Speeder neuerer Bauart kommen dieser Geschwindigkeit

a) vom Computer zur Floppy

Computerroutine				Floppyroutine			
	LDA	255	;Datenrichtung		LDX	16	
	STA	$DD03	;auf Ausgang	L1	LDA	$1800	;auf ATN
	LDA	11			BPL	L1	;warten
	STA	$DD00	;ATN setzen		STX	$1800	;DATA setzen
	LDX	03			NOP		;auf Computer
L1	LDA	$DD00	;auf DATA		NOP		;warten
	BPL	L1	;warten		LDX	00	
	STX	$DD00	;ATN rücksetzen		STX	$1800	;DATA rücksetzen
L2	LDA	Puffer,Y	;Datenbyte holen		NOP		;auf Computer
	LDX	11			NOP		;warten
	STX	$DD00	;ATN setzen	L2	LDA	$1800	;auf ATN
	STA	$DD01	;Datenbyte senden		BPL	L2	;warten
	LDX	03			LDA	$1801	;Datenbyte holen
	STX	$DD00	;ATN rücksetzen		STA	Puffer,Y	;und sichern
	INY				NOP		
	BNE	L2	;nächstes Byte		NOP		
	LDX	00	;Datenrichtung		INY		
	STX	$DD03	;auf Eingang		BNE	L2	;nächstes Byte
	RTS		;Datenbyte holen		LDX	15	
					STX	$1800	;ATN rücksetzen
					RTS		

b) Von der Floppy zum Computer

Computerroutine				Floppyroutine			
	LDX	11			LDA	255	;Datenrichtung
	STX	$DD00	;ATN setzen		STA	$1803	;auf Ausgang
	LDX	03			LDX	16	
L1	LDA	$DD00	;auf DATA	L1	LDA	$1800	;auf ATN
	BPL	L1	;warten		BPL	L1	;warten
	STX	$DD00	;ATN rücksetzen		STX	$1800	;DATA setzen
L2	LDA	11		L2	LDA	Puffer,Y	;Datenbyte holen
	STA	$DD00	;ATN setzen		NOP		
	NOP		;warten auf		NOP		;auf Computer
	NOP		;Floppy		NOP		;warten
	LDX	03		L3	LDX	$1800	;auf ATN
	STX	$DD00	;ATN rücksetzen		BPL	L3	;warten
	LDA	$DD01	;Datenbyte holen		STA	$1801	;Datenbyte senden
	STA	Puffer,Y	;und speichern		INY		
	NOP				BNE	L2	;nächstes Byte
	NOP				LDX	15	
	INY				STX	$1800	;ATN rücksetzen
	BNE	L2	;nächstes Byte		LDX	00	;Datenrichtung
	RTS				STX	$1803	;auf Eingang
					RTS		

Tabelle 3. Fast-Routinen zur Parallel-Übertragung. Der Datenpuffer sowie die Tabellen dürfen nicht Page-übergreifend im Speicher liegen, da sonst bei X/Y-Indizierung ein Taktzyklus mehr benötigt wird.

a) Vom Schreib-/Lese-Kopf zum Computer

Computerroutine

```
     LDA  #<Adresse  ;Pointer für
     STA  $FA        ;Adresse
     STA  $FC        ;im Speicher
     LDX  #>Adresse  ;setzen
     STX  $FB

     INX
     STX  $FD
     LDY  #00
L1   BIT  $DD00      ;warten auf
     BPL  L1         ;gesetztes DATA
     LDA  $DD01      ;Datenbyte holen
     STA  ($FA),Y    ;und sichern
     INY
L2   BIT  $DD00      ;warten auf
     BMI  L2         ;gelöschtes DATA
     LDA  $DD01      ;Datenbyte holen
     STA  ($FA),Y    ;und sichern
     INY
     BNE  L1         ;nächste Bytes
L3   BIT  $DD00      ;warten auf
     BPL  L3         ;gesetztes DATA
     LDA  $DD01      ;Datenbyte holen
     STA  ($FC),Y    ;und sichern
     INY
L4   BIT  $DD00      ;warten auf
     BMI  L4         ;gelöschtes DATA
     LDA  $DD01      ;Datenbyte holen
     STA  ($FC),Y    ;und sichern
     INY
     CPY  68         ;=324-256 Bytes
     BNE  L3
     RTS
```

Floppyroutine

```
     LDA  255        ;Datenrichtung
     STA  $1803      ;auf Ausgang
     STA  $1800      ;DATA zurücksetzen
     LDY  162        ;=320/2 Bytes
     JSR  $F50A      ;Auf Datenblock-SYNC
                      warten
L1   BVC  L1         ;Warten auf
     CLV             ;Byte-Ready
     LDA  $1C01      ;Datenbyte holen
     STA  $1801      ;und übertragen
     INC  $DD00      ;DATA setzen
L2   BVC  L2         ;Warten auf
     CLV             ;Byte-Ready
     LDA  $1C01      ;Datenbyte holen
     STA  $1801      ;und übertragen
     DEC  $DD00      ;DATA zurücksetzen
     DEY
     BNE  L1         ;nächste Bytes
     LDA  #00        ;Datenrichtung
     STA  $1803      ;auf Eingang
     RTS
```

b) Vom Computer zum Schreib-/Lese-Kopf

Computerroutine

```
     LDA  #<Adresse  ;Pointer für
     STA  $FA        ;Adresse
     STA  $FC        ;im Speicher
     LDX  #>Adresse  ;setzen
     STX  $FB
     INX
     STX  $FD
     LDA  255        ;Datenrichtung
     STA  $DD03      ;auf Ausgang
     LDY  #00
L1   BIT  $DD00      ;warten auf
     BPL  L1         ;gesetztes DATA
     LDA  ($FA),Y    ;Datenbyte holen
     STA  $DD01      ;und übertragen
     INY  L1
L2   BIT  $DD00      ;warten auf
     BMI  L2         ;gelöschtes DATA
     LDA  ($FA),Y    ;Datenbyte holen
     STA  $DD01      ;und übertragen
     INY
     BNE  L1         ;nächste Bytes
L3   BIT  $DD00      ;warten auf
     BPL  L3         ;gesetztes DATA
     LDA  ($FC),Y    ;Datenbyte holen
     STA  $DD01      ;und übertragen
     INY
L4   BIT  $DD00      ;warten auf
     BMI  L4         ;gelöschtes DATA
     LDA  ($FC),Y    ;Datenbyte holen
     STA  $DD01      ;und übertragen
     INY
     CPY  68         ;=324-256 Bytes
     BNE  L3
     LDA  #00        ;Datenrichtung
     STA  $DD03      ;auf Eingang
     RTS
```

Floppyroutine

```
     LDA  00         ;DATA setzen
     STA  $1800
     JSR  $F510      ;Sektorheader suchen
     LDY  162        ;=320/2 Bytes
     LDA  255        ;Kopfregister auf
     STA  $1C03      ;Ausgang stellen
     LDA  $1C0C      ;Elektronik
     AND  #$1F       ;auf
     ORA  #$C0       ;Schreiben
     STA  $1C0C      ;umschalten
     LDA  #$FF       ;Wert für SYNC
     LDX  #05        ;5 mal schreiben
     STA  $1C01      ;Byte zum Kopf geben
     CLV             ;Byte-Ready
     BVC  L1         ;Warten auf Schreiben
     CLV
     DEX
     BNE  L1         ;5*$FF schreiben
L2   LDA  $1801      ;Datenbyte holen
     DEC  $1800      ;DATA zurücksetzen
     STA  $1C01      ;Datenbyte auf
L3   BVC  L3         ;Diskette
     CLV             ;schreiben
     LDA  $1801      ;Datenbyte holen
     INC  $1800      ;DATA setzen
     STA  $1C01      ;Datenbyte auf
L4   BVC  L4         ;Diskette
     CLV             ;schreiben
     DEY
     BNE  L2         ;nächste Bytes
     LDA  #15
     STA  $1800      ;DATA zurücksetzen
     JSR  $FE00      ;auf Lesen umschalten
     RTS
```

Tabelle 4. Übertragung zwischen Computer und Schreib-/Lese-Kopf

sehr nahe. Besonders Kopierprogramme profitieren davon, einen Track komplett lesen zu können. In Tabelle 4 sehen Sie Routinen, die einen Sektor in codierter Form (324 Byte inklusive zwei Abschlußbytes, Datenblockkennzeichen ($07) und Checksumme) direkt vom/zum Schreib-/Lese-Kopf übertragen.

Damit sind wir am Ende der Datenübertragung angelangt und wollen uns den floppyinternen Verbesserungen zuwenden. Als erstes ist die GCR-Codierung/Decodierung zu nennen, die vom Floopy-DOS umständlich und damit langsam vorgenommen wird. Was hat es mit dieser Codierung auf sich? Nun, um den Anfang eines Sektors auf der Diskette zu finden, schreibt die Floppy vor jeden Block eine sogenannte SYNC-Markierung. Dabei handelt es sich um 5 aufeinanderfolgende $FF-Bytes, also um insgesamt 40 gesetzte Bit. Die Schreib-/Lese-Elektronik erkennt, wenn mehr als 10 Bit aufeinanderfolgen und löst daraufhin das SYNC-Signal aus. Der Nachteil dieser Methode besteht darin, daß normale auf der Diskette gespeicherte Binär-Bytes mit einer SYNC-Markierung verwechselt werden könnten:

```
Binärbytes:...—$3F—$F8—...
          00111111/11111000
          = 11 folgende '1'-Bits
```

Sie sehen, daß nicht nur gespeicherte $FF-Bytes, sondern auch andere Kombinationen für Verwirrung sorgen würden. Um dies zu vermeiden, führte Commodore das sogenannte GCR-(Group-Code-Recording) Verfahren ein, dessen Idee darin besteht, ein Binär-Byte in zwei Hälften zu zerlegen und diese beiden Halbbytes nach einer Tabelle in 5-Bit-GCR-Bytes umzuwandeln. Da auf diese Weise aus jedem Binär-Byte 10 Bit erzeugt werden, muß man die so entstandenen Bytes vor der Speicherung zerlegen, da ein Diskettenbyte natürlich nur 8 Bit enthalten kann:

```
1.GCR-Byte   2.GCR-Byte   3.GCR-Byte   4.GCR-Byte
1111111111/  2222222222/  3333333333/  4444444444
werden zu
11111111/11222222/22223333/33333344/44444444
1.,      2.,      3.,      4.,      5. Diskettenbyte
```

Sie erkennen, daß jeweils 4 Binärbytes in 5 Diskettenbytes umgewandelt werden, so daß jeder Sektor statt 256 im GCR-Format 320 Byte enthält. Hier nun die Tabelle zur Umrechnung in den GCR-Code:

Dezimal	Binär	GCR-Code	Dezimal	Binär	GCR-Code
0	0000	01010	8	1000	01001
1	0001	01011	9	1001	11001
2	0010	10010	10	1010	11010
3	0011	10011	11	1011	11011
4	0100	01110	12	1100	01101
5	0101	01111	13	1101	11101
6	0110	10110	14	1110	11110
7	0111	10111	15	1111	10101

Nach maximal 4 Bit mit einer »1« folgt wieder ein »0«-Bit, so daß insgesamt nicht mehr als 6 »1«-Bits nacheinanderliegen können. Die Verwechslung mit einer SYNC-Markierung ist daher nicht möglich. Die GCR-Codierung hat jedoch den entscheidenen Nachteil, daß die Routinen zur Codierung beziehungsweise Decodierung umständlich und daher ziemlich zeitraubend sind. Daher ist es selbst bei der Parallelübertragung nicht möglich, die Diskettenbytes zu lesen, zu decodieren und zu übertragen, bevor das nächste Byte anliegt. Die einzige Lösung besteht in der hardware-gesteuerten Decodierung, wie sie beispielsweise vom Speeder Professional-DOS vorgenommen wird. Andere Systeme besitzen zusätzlichen RAM-Speicher, in den ein kompletter Track eingelesen und anhand von längeren Tabellen schneller decodiert werden kann.

Weil die 320 Byte eines Sektors nicht in einen RAM-Puffer der Floppy passen, besitzt diese den sogenannten GCR-Zusatzpuffer von $01BB bis $01FF, der die restlichen 64 Byte aufnimmt. Die Orginal-Routine zur GCR-Decodie-

```
GCR-Daten stehen bei $0300-$03FF und $01BA-$01FF.
     LDY  00        ;Pufferzeiger
     STY  $30       ;auf Anfang setzen
     STY  $34
     LDY  03
     STY  $31
     STY  $36       ;5 GCR-Byte in 4 Binärbyte
     JSR  $F7E6     ;umwandeln
     LDA  $52       ;erstes Binärbyte als Kennzeichen
     STA  $38       ;für Blockheader merken
     LDX  02
L1   LDA  $53,X     ;die anderen 3 Byte
     STA  $0300,X   ;in Puffer zurückschreiben
     DEX
     BPL  L1
Es folgt die Umwandlungsroutine für je 5 GCR-Byte
L2   LDA  ($30),Y   ;erstes Datenbyte holen
     LSR  A
     LSR  A         ;oberes GCR-Halbbyte isolieren
     LSR  A
     TAX            ;und merken
     LDA  ($30),Y   ;erstes Datenbyte nochmal holen
     AND  07        ;Bits 0-2 für unteres GCR-Halbbyte
     STA  $52       ;merken
     INY            ;Hauptpuffer komplett decodiert ?
     BNE  L4        ;Nein
     LDA  01        ;Ja, Zeiger auf GCR-
     STA  $31       ;Zusatzpuffer
     LDY  186
L4   LDA  ($30),Y   ;zweites Datenbyte holen
     ASL  A         ;höchste 2 Bits in voriges,
     ROL  $52       ;unteres GCR-Halbbyte
     ASL  A         ;als Bits 0-2
     ROL  $52       ;einblenden
     LDA  $F8A0,X   ;Oberes GCR-Halbbyte umwandeln
     LDX  $52       ;und unteres GCR-Halbbyte
     ORA  $FAC0,X   ;umgewandelt einblenden
     LDX  $36       ;Position im Datenpuffer holen
     STA  $0300,X   ;Binärbyte abspeichern
     LDA  ($30),Y   ;zweites Datenbyte nochmal holen
     LSR  A         ;Bits 1-5 als oberes GCR-Halbbyte
     AND  31        ;isolieren
     TAX            ;und merken
     INY
     LDA  ($30),Y   ;drittes Datenbyte holen
     ROR  A         ;Bits 4-7 und Bit 0
     LSR  A         ;des zweiten Datenbytes
     LSR  A         ;als unteres GCR-Halbbyte
     LSR  A         ;isolieren
     STA  $53       ;und merken
     LDA  $F8A0,X   ;Oberes GCR-Halbbyte umwandeln
     LDX  $53       ;und unteres GCR-Halbbyte
     ORA  $FAC0,X   ;umgewandelt einblenden
     LDX  $36       ;Position im Datenpuffer holen
     INX            ;und erhöhen
     BEQ  L3        ;Decodierung fertig
     STA  $0300,X   ;Binärbyte speichern
     LDA  ($30),Y   ;Drittes Datenbyte nochmal holen
     AND  15        ;Bits 0-3 isolieren und merken als
     STA  $54       ;Bits 1-4 des oberen GCR-Halbbytes
     INY
     LDA  ($30),Y   ;Viertes Datenbyte holen
     STA  $55       ;merken
     ASL  A         ;7.Bit als Bit 0 des oberen
     ROL  $54       ;GCR-Halbbytes setzen
     LSR  A         ;Bits 2-6 als Bits 0-4 des unteren
     LSR  A         ;GCR-Halbbytes merken
     LSR  A
     TAX
     LDA  $F8C0,X   ;Unteres GCR-Halbbyte umwan-
                    deln
     LDX  $54       ;und oberes GCR-Halbbyte
     ORA  $FAA0,X   ;umgewandelt einblenden
     LDX  $36       ;Position im Datenpuffer holen
     STA  $0302,X   ;Binärbyte abspeichern
     INY
     LDA  ($30),Y   ;fünftes Datenbyte holen
     ASL  A         ;Bits 5-7 des fünften mit
     ROL  $55       ;den Bits 0-1 des vierten
     ASL  A         ;Datenbytes als oberes
     ROL  $55       ;GCR-Halbbyte isolieren
     ASL  A
     ROL  $55
     ROL  A
     AND  31
     TAX
     LDA  $F8A0,X   ;Oberes GCR-Halbbyte umwandeln
     STA  $55       ;und dieses Binärhalbbyte merken
     LDA  ($30),Y   ;fünftes Datenbyte nochmal holen
     AND  31        ;Bits 0-4 als unteres GCR-Halbbyte
     TAX            ;isolieren, umwandeln
     LDA  $F8C0,X   ;und in oberes
     ORA  $55       ;Binärhalbbyte einblenden
     LDX  $36       ;Pufferzeiger holen
     STA  $0303,X   ;Binärbyte speichern
     INY
     TXA
     CLC            ;Pufferzeiger um 4 Byte
     ADC  04        ;erhöhen
     STA  $36
     JMP  L2        ;nächste 5 Byte decodieren
L3   RTS
```

Tabelle 5. Eine Routine zur schnellen GCR-Decodierung des Puffers 0

rung steht von $F5F2-$F690 im Floppy-DOS und ist zum Beispiel im Buch »Die Floppy 1541« von Karsten Schramm aus dem Markt und Technik Verlag kommentiert. Die Unterprogramm-Aufrufe dienen dazu, jeweils 5 GCR-Byte in 4 Binär-Byte umzuwandeln. Die Pufferadresse muß dazu in den Speicherstellen $30 und $31 abgelegt sein. Die Unterroutine legt die 4 erzeugten Binär-Byte in dem Speicher von $52-$55 ab, so daß diese von dort wiederin den Puffer zurückkopiert werden müssen. Da das DOS im ROM steht und sich nicht selbst modifizieren kann, ist dieses Verfahren das einzig mögliche, da alle fünf RAM-Puffer verwendet werden können. Unser Floppy-Programm liegt aber im RAM und braucht den Zwischenspeicher nicht, sondern es kann die Binär-Bytes direkt in den Puffer zurückschreiben. Leider kann ich an dieser Stelle die originale Umwandlungsroutine aus Platzgründen nicht darstellen.

In Tabelle 5 sehen Sie eine Routine, die die GCR-Bytes wesentlich schneller decodiert als das Floppy-DOS. Die Daten müssen vorher im Puffer von $0300-$03FF und im GCR-Zusatzpuffer stehen. Um auch andere Puffer decodieren zu können, braucht man in der Routine nur alle Adressen, die sich auf den Bereich von $0300-$03FF beziehen, zu ändern. Aus Platzgründen kann ich eine Routine zur schnellen GCR-Codierung nicht mehr vorstellen. Diese läßt sich aber nach den gleichen Prinzipien entwickeln. Die entsprechende Original-Routine im Floppy-DOS steht von $F78F-$F7E5.

Neue Ordnung

Neben der GCR-Codierung ist es besonders zeitaufwendig, einen bestimmten Sektor auf der Diskette zu suchen. Das DOS speichert die einzelnen Sektoren einer Datei versetzt mit einem Zwischenraum von je neun bis zehn Blöcken ab, das heißt nach Sektor 1 folgt nicht Sektor 2 sondern Sektor 11, dann Sektor 2, 12, und so weiter. Der Grund hierfür besteht in der normalerweise langsamen Datenübertragung über den seriellen Bus. In der Praxis hat sich gezeigt, daß man pro Sektor zirka eine halbe Diskettenumdrehung benötigt. Bei der schnellen seriellen Übertragung ist es jedoch möglich, vier Blöcke pro Umdrehung einzulesen, so daß eine Versetzung um nur fünf Sektoren besser wäre. Die Parallelübertragung ermöglicht es sogar, einen Track pro Umdrehung einzulesen, so daß überhaupt keine Versetzung mehr erforderlich ist. Deshalb speichern Floppy-Speeder meist in der tatsächlichen Reihenfolge ab, das heißt nach Sektor 1 folgt Sektor 2, 3, und so weiter. Um versetzte Tracks schneller lesen zu können, muß man sich daher eines Tricks bedienen: Während einer Umdrehung werden die Header aller Blöcke gelesen, so daß die Struktur einer Datei erfaßt werden kann. Man liest die Blöcke dann nicht in der eigentlichen Reihenfolge ein, sondern immer den gerade »vorbeikommenden« Block, sofern er zu der Datei gehört. Dieses Prinzip wird auch von EXOS verwendet und stellt den Hauptgrund für die Beschleunigung dar. Die Übertragungsroutinen sind denen von Hypra-Load ähnlich, dennoch kann man die Leistungsfähigkeit beider Systeme nicht vergleichen.

Das Hauptproblem besteht darin, festzustellen, um welchen Sektor es sich jeweils handet. Die Informationen hierfür stehen in dem Blockheader (Vorspann), der nach der Sync-Markierung folgende Daten enthält: Headerkennzeichen ($08), Checksummer aller Headerbytes, Sektornummer, Tracknummer, zwei ID-Bytes sowie zwei Abschlußbytes. Das Headerkennzeichen dient dazu, den Vorspann vom eigentlichen Datenblock zu unterscheiden, der ebenfalls durch eine Sync-Markierung eingeleitet wird und als Kennzeichen die Zahl $07 aufweist. Die beiden Abschlußbyte sind erforderlich, um eine durch 4 teilbare Zahl an Headerbyte (8) zu erhalten, da ja immer 4 Binär-Byte in 5 GCR-Byte umgewandelt werden. Zwischen Header und Datenblock liegt eine Lücke von 8 Byte, um der Elektronik die Zeit

```
      LDX   42        ;42 Leseversuche
L1    LDA   208       ;Timer laden (ca. 53 ms)
      STA   $1805     ;und starten
L5    BIT   $1805     ;Timer abgelaufen ?
      BPL   L6        ;Ja, Ende
      BIT   $1C00     ;SYNC gefunden ?
      BMI   L5        ;Nein, weiter warten
      LDA   $1C01     ;Ja, Kopf initialisieren
      CLV             ;Elektronik ready
L2    BVC   L2        ;warten auf Disk-Byte
      CLV
      LDA   $1C01     ;Byte holen
      CMP   82        ;Kennzeichen für Header?
      BEQ   OK        ;Ja
      DEX             ;nächster Leseversuch
      BNE   L1
L6    RTS             ;Kein Sektor gefunden
OK    LDX   03        ;2 Byte überlesen
L3    BVC   L3        ;warten auf Disk-Byte
      CLV
      LDA   $1C01     ;Byte holen
      DEX             ;(Diskbytes 2 und 3
      BNE   L3        ;überlesen)
      TAX             ;Diskbyte 4 merken
      PHP             ;Statusregister retten
      LSR   A         ;Bits 2 - 7 in Position
      LSR   A         ;0 - 5 schieben
      AND   31        ;Bits 0-4 isolieren
      TAX             ;unteres GCR-Halbbyte in
      LDA   $F8C0,X   ;Binärbyte umwandeln
      PLP             ;Status holen
      BPL   L4        ;6.GCR-Bit prüfen
      ORA   16        ;5.Binärbit=1, d.h. setzen
L4    STA   SEKTOR    ;decodierte Sektornummer
      RTS
```

Tabelle 6. Decodierung der Sektornummer im Blockheader

für das Umschalten auf Schreiben zu geben. Leider ist diese Zeit aber mit maximal 256 Mikrosekunden viel zu kurz, um den Header zu decodieren, bevor die Daten-Byte anliegen. Das DOS löst dieses Problem, indem es einen »Soll«-Header im RAM codiert und diese Daten byteweise mit dem jeweiligen Blockheader vergleicht. Stimmen alle 8 Byte überein, ist der Block offenbar gefunden worden, worauf die Daten-Byte gelesen werden können. Bei der Codierung wird immer die gesuchte Sektornummer eingesetzt. Man sieht sofort, daß dieses Verfahren aber nicht funktioniert, wenn man keinen bestimmten Block sucht, da die Sektornummer ja nicht bekannt ist. Diese wollen wir ja gerade aus dem Header erfahren! Um das anfangs noch als unmöglich bezeichnete nachträgliche Decodieren doch zu realisieren, muß man sich klarmachen, daß jeder Sektor eines Tracks außer der Sektornummer und der Checksumme völlig identischen Inhalt besitzt. Wenn man davon ausgeht, daß die Checksumme stimmt, brauchte man eigentlich nur jeweils die Sektornummer und nicht den ganzen Header zu decodieren. Dazu müssen wir zunächst einmal sehen, wo wir diese finden:

	Headerkennzeichen	Checksumme	Sektornummer	Tracknummer
GCR-Code	1111111111	2222222222	3333333333	4444444444
Diskette	11111111 /	11222222 /	22223333 /	33333344 / 44444444

```
      LDX   00        ;Wert für Innenbewegung
      LDA   TRACK     ;Zieltracknummer holen
      SEC
      SBC   $22       ;minus letzter Track
      BCC   L1        ;Unterlauf, dann nach innen
      LDX   01        ;Wert für Außenbewegung
L1    STX   L2+1      ;Wert für Innen/Außen speichern
      ASL   A         ;2 Schritte pro Track
      TAX             ;als Zähler setzen
L2    LDA   00        ;00 (innen) oder 01 (außen)
      EOR             $1C00 ;invertieren
      SEC             ;um 1 Bit nach links, dabei
      ROL   A         ;Bit 0 =1
      AND   03        ;Bits 0 und 1 isolieren
      EOR   $1C00     ;invertieren
      STA   $1C00     ;= Schrittmotorbewegung
      LDA   150       ;warten, bis Mechanik
      STA   $1805     ;Bewegung ausgeführt hat
L3    LDA   $1805     ;(wenn Timer
      BMI   L3        ;abgelaufen ist)
      DEX
      BNE   L2        ;nächster Schritt
      RTS
```

		Innenbewegung (4 Schritte)		Außenbewegung (4 Schritte)	
Befehl		Akku	$1C00	Akku	$1C00
LDA	00/01	00000000	??????10	00000001	??????10
EOR	$1C00	??????10	??????10	??????11	??????10
SEC		??????10	??????10	??????11	??????10
ROL	A	?????101	??????10	?????111	??????10
AND	03	?????001	??????10	?????011	??????10
EOR	$1C00	??????11	??????10	??????01	??????10
STA	$1C00	??????11	??????11	??????01	??????01
LDA	00/01	00000000	??????11	00000001	??????01
EOR	$1C00	??????11	??????11	??????00	??????01
SEC		??????11	??????11	??????00	??????01
ROL	A	?????111	??????11	?????001	??????01
AND	03	?????011	??????11	?????001	??????01
EOR	$1C00	??????00	??????11	??????00	??????01
STA	$1C00	??????00	??????00	??????00	??????00
LDA	00/01	00000000	??????00	00000001	??????00
EOR	$1C00	??????00	??????00	??????01	??????00
SEC		??????00	??????00	??????01	??????00
ROL	A	?????001	??????00	?????011	??????00
AND	03	?????001	??????00	?????011	??????00
EOR	$1C00	??????01	??????00	??????11	??????00
STA	$1C00	??????01	??????01	??????11	??????11
LDA	00/01	00000000	??????01	00000001	??????11
EOR	$1C00	??????01	??????01	??????10	??????11
SEC		??????01	??????01	??????10	??????11
ROL	A	?????011	??????01	?????101	??????11
AND	03	?????011	??????01	?????001	??????11
EOR	$1C00	??????10	??????01	??????10	??????11
STA	$1C00	??????10	??????10	??????10	??????10

und so weiter ...

Tabelle 7. Schritt-Motorsteuerung mit Beispiel

Unsere Sektornummer finden wir also in zwei Disketten-Byte aufgeteilt: Die oberen 4 GCR-Bit liegen an der Position 0 bis 3 im dritten Byte, die unteren sechs im vierten Byte an der Position 2 bis 7. Offenbar müssen 2 Byte gelesen werden, um die Nummer zu erhalten. In der Praxis benötigt man aber nur das vierte Byte. Da die Sektornummer maximal den Wert 20 (Binär: 00010100) annehmen kann, brauchen nur die unteren fünf Binärbit betrachtet zu werden. Zunächst wollen wir aber annehmen, daß die Sektornummer nur im Bereich von 0 bis 15 liegt, wir also sogar nur die untersten 4 Bit benötigen. In diesem Fall brauchen wir natürlich auch nur das untere GCR-Halbbyte anzusehen, die GCR-Bit 0 bis 4. Diese liegen aber komplett im vierten Diskettenbyte. Falls nun die Sektornummer im Bereich von 16-20 liegt, ist das vierte Binär-Bit eins, die Binär-Bit 5-7 sind jedoch immer null. Das obere Binär-Halb-Byte kann also nur die Werte 1 oder 0 annehmen. Wenn wir nun die GCR-Tabelle ansehen, erkennen wir, daß sich die GCR-Halbbyte für 0 und 1 bereits im untersten Bit unterscheiden: Während die Null durch die Kombination 01010 dargestellt wird, wird die Eins in das GCR-Halbbyte 01011 umgewandelt. Da sich das unterste Bit des zweiten GCR-Halbbytes jedoch noch im vierten Disketten-Byte des Headers befindet, können wir auf das dritte Diskettenbyte völlig verzichten und brauchen damit nur 1 Byte zu lesen. Die Routine in Tabelle 6 erledigt diese Aufgabe so schnell, daß die Sektornummer zur Verfügung steht, noch bevor der Header vorbei ist. Man hat sogar noch Zeit, auf Schreiben umzuschalten. Mit dieser Routine lassen sich sehr effektive Speeder programmieren, da man pro Umdrehung der Diskette wesentlich mehr Blöcke lesen kann.

Step by step

Als letzten Themenbereich möchte ich die Schrittmotor-Steuerung der 1541 ansprechen. Hier sind die DOS-Routinen wirklich katastrophal programmiert, so gibt es eine völlig überflüssige Routine zum Abbremsen des Motors! Pro Track müssen zwei Schritte ausgeführt werden, wodurch man auch die sogenannten Halftracks ansteuern kann, indem man nur einen Schritt ausführt. Wie langsam der Motor durch das DOS angesteuert wird, belegt das folgende Beispiel, das den Kopf von Spur 1 auf Spur 35 fährt:

```
10 OPEN 1,8,15:OPEN 2,8,2,"#"
20 PRINT #1,"U1 2 0 1 0"
30 PRINT #1,"U1 2 0 35 0"
40 CLOSE1:CLOSE2
```

Die Regel, daß sich der Kopf dann schnell genug bewegt, wenn man ihn nicht hört, trifft hier offensichtlich (»offenhörbar«) nicht zu. Man kann eher den Eindruck gewinnen, daß ein riesiges Mahlwerk seine Dienste verrichtet.

Für die Steuerung des Schrittmotors sind die Bits 0 und 1 des Registers 0 der CIA 2 (ab $1C00) zuständig. Die Bewegung wird durch die Folge der Inhalte dieser Bits gesteuert: Durch die Folge

...—00—01—10—11—00—01—...

wird der Kopf nach innen bewegt, während er durch die Folge

...—00—11—10—01—00—11—...

nach außen wandert. Durch jede Änderung wird jeweils ein Schritt ausgeführt. Nach jeder Steuerung muß man einen Moment warten, da die mechanische Bewegung dem Tempo der Maschinenprogrammierung natürlich nicht folgen kann. Das DOS ist jedoch übervorsichtig und schont übermäßig den Motor, jedoch nicht die Geduld des Floppy-Besitzers. In Tabelle 7 sehen Sie eine extrem schnelle Steuer-Routine, die bei einer Bewegung nur noch ein Summen hören läßt. Besonders bei gefüllten Disketten, wo ständig längere Bewegungen des Kopfes erforderlich sind, bringt deren Anwendung erhebliche Vorteile. Da die Funktion auf den ersten Blick wohl nicht gerade einleuchtend ist, wird diese durch ein Beispiel in Tabelle 7 schrittweise dokumentiert.

Wir sind nun am Ende dieser kleinen Exkursion angekommen. Mit den gezeigten Routinen müßten Sie in der Lage sein, die 1541 »auf Trab« zu bringen. Durch die Programme EXOS und Master-Copy (64'er Magazin) sowie Master-Copy parallel in diesem Sonderheft, wurde ja bewiesen, was man aus dieser Floppy herausholen kann.

(Frank Riemenschneider/ap)

Von Programmschutz bis zum farbigen Listing

Erst Diskmonitore machen es möglich, alle Arten der Diskettenmanipulation sinnvoll und effektiv zu nutzen. Dies soll hier anhand von vielen Beispielen verdeutlicht werden.

Spätestens nach dem ersten Versuch, eine geschützte Datei zu löschen oder eine versehentlich gelöschte zu »restaurieren«, werden Sie sich wahrscheinlich gefragt haben, wie man das entprechende Häkchen im Inhaltsverzeichnis denn anbringen oder wieder aufheben kann, oder ob es noch eine Rettungsmöglichkeit für die Datei gibt. Die Antwort ist einfach:

Mit einem Diskmonitor! Dieser bietet problemlosen und einfachen Zugriff auf nahezu alle (!) Daten einer Diskette, auch auf die in diesem Fall wirksamen Dateiattribute. Anhand von vielen einfachen und praktischen Beispielen wollen wir nun im Rahmen dieses Artikels die Anwendung dieser speziellen Monitore kennenlernen. Dazu ist nicht mehr nötig, als eine Floppy und ein beliebiger passender Standard-Diskmonitor. Gute und leistungsfähige Programme dieser Art für den C 128 und den C 64, den »TOP-FLOP« und den »Disc-Scanner 40«, finden Sie ebenfalls in diesem Sonderheft. Durch die weitgehende Übereinstimmung der Formate der Floppies 1541, 1551, 1570 und 1571, sind kaum Differenzierungen nötig, im Zweifelsfalle beziehen sich die Angaben jedoch auf die am weitesten verbreitete Floppy 1541.

Was leistet ein Diskmonitor?

Die Hauptaufgabe eines Diskmonitors ist die exakte Darstellung der Inhalte der einzelnen Sektoren (Informationen über Aufbau und Organisation einer Diskette entnehmen Sie bitte dem Beitrag »In die Geheimnisse der Floppy eingetaucht«, der ebenfalls in diesem Heft zu finden ist). Diese Darstellung erfolgt standardmäßig in zwei verschiedenen Codierungen, nämlich als Hexadezimal-Dump sowie als ASCII-Dump.

Der Hexdump wird zur Ausgabe der einzelnen Byte-Werte benutzt, da alle Zahlen zwischen 0 und 255 (also alle möglichen Byte-Werte) nur zwei Ziffern benötigen. Somit ist eine übersichtliche und gleichmäßige Tabelle möglich. Die Umsetzung der Bytes in Buchstaben und Zeichen hilft dem Anwender dabei, Texte innerhalb der Sektoren zu erkennen und zu lesen, ohne ständig die ASCII-Tabelle wälzen zu müssen. Steuerzeichen sowie unwichtige Grafikzeichen erscheinen bei den meisten Monitoren als einfacher Punkt ».«.

Diese beiden Formate kennen Sie vielleicht schon von Speichermonitoren wie dem schon legendären SMON aus dem 64'er-Magazin oder dem TEDMON des C 128. Die Floppymonitore zeigen jedoch keine bestimmten Teile des Speichers, sondern von Teilbereichen von Disketten, normalerweise Teile von Dateien an. Zur Veranschaulichung dient Listing 1, ein Programm, das einen beliebigen Sektor wahlweise auf Bildschirm oder Drucker ausgibt, es funktioniert dank der Verwendung nur einfachster Befehle übrigens auf jedem Commodore-Heimcomputer mit 40 oder mehr Zeichen pro Zeile. Interessant ist auch die anschauliche Umrechnung von Dezimal- in Hexadezimalzahlen ab Zeile 600, die Sie in eigene Programme übernehmen können. Bild 1 zeigt einen typischen Ausdruck des ersten Directory-Blockes, die Zahl am Anfang jeder Zeile ist die Nummer des ersten Byte.

Weitere Funktionen

Die Funktion eines Diskmonitors beschränkt sich allerdings nicht allein auf die Ausgabe von Sektoren, was zwar ebenfalls sehr aufschlußreich, jedoch auf die Dauer doch nicht so ganz befriedigend sein kann. Erst das Verändern der Daten auf der Diskette macht den Umgang mit dem Diskmonitor zum interessanten – aber auch risikoreichen – Abenteuer. Diese Funktion wollen wir uns mit Hilfe des kleinen Programms »LITTLE ED« (Listing 2) verdeutlichen.

```
00 : 12 04 82 11 00 45 41 53    ....EAS
08 : 59 2D 44 55 4D 50 A0 A0    ........
10 : A0 A0 A0 A0 A0 00 00 00         ...
18 : 00 00 00 00 00 00 05 00    ........
20 : 00 00 82 13 00 54 4F 50    ....TOP
28 : 2D 46 4C 4F 50 2E 53 55    -FLOP.SU
30 : 42 A0 A0 A0 A0 00 00 00    B    ...
38 : 00 00 00 00 00 00 03 00    ........
40 : 00 00 00 13 01 45 41 53    .....EAS
48 : 59 2D 44 55 4D 50 A0 A0    ........
50 : A0 A0 A0 A0 A0 00 00 00         ...
58 : 00 00 00 00 00 00 05 00    ........
60 : 00 00 82 14 03 4C 49 54    ....LIT
68 : 54 4C 45 20 45 44 A0 A0    ........
70 : A0 A0 A0 A0 A0 00 00 00         ...
78 : 00 00 00 00 00 00 03 00    ........
80 : 00 00 82 13 0E 42 49 54    ....BIT
88 : 2D 53 55 43 48 45 52 A0    ........
90 : A0 A0 A0 A0 A0 00 00 00         ...
98 : 00 00 00 00 00 00 03 00    ........
A0 : 00 00 82 14 01 44 49 52    ....DIR
A8 : 20 4B 4F 4D 50 4C 45 54     KOMPLET
B0 : 54 A0 A0 A0 A0 00 00 00    T    ...
B8 : 00 00 00 00 00 00 04 00    ........
C0 : 00 00 82 14 04 42 43 2D    ....BC-
C8 : 53 55 43 48 45 52 A0 A0    ........
D0 : A0 A0 A0 A0 A0 00 00 00         ...
D8 : 00 00 00 00 00 00 06 00    ........
E0 : 00 00 82 13 03 44 41 54    ....DAT
E8 : 45 49 53 43 48 4E 55 45    EISCHNUE
F0 : 46 46 4C 45 52 00 00 00    FFLER...
F8 : 00 00 00 00 00 00 09 00    ........
```

Bild 1. Beispiel eines Ausdrucks mit »EASY-DUMP«

Auch er soll mehr anschauliches Beispiel als praxisgerechte Hilfe sein. Seien Sie aber trotzdem vorsichtig, die mit seiner Hilfe veränderten Daten sind ebenso unwiederbringlich überschrieben wie bei jedem anderen Programm dieser Art.

Überhaupt empfiehlt es sich, vor irgendwelchen Veränderungen zuerst eine Hardcopy des ursprünglichen Sektors zu machen, damit Sie im Notfall den alten Zustand wieder herstellen können. Bei Top-Flop wird übrigens, wie bei vielen anderen Diskmonitoren, der Sektor zunächst nur im Speicher editiert. Zurückgeschrieben wird er erst durch einen besonderen Befehl.

Sicher ist sicher

Ein weiterer Befehl, der in fast jedem Diskmonitor verfügbar ist, ist das Laden beziehungsweise Darstellen des logisch nächsten Sektors einer Datei. Unterschieden werden muß hier zwischen dem physikalisch nächsten und logisch nächsten Sektor: Der physikalisch nächste ist der direkt folgende. Also bei Sektor 18/01 beispielsweise 18/02. Der logisch nächste ist der, in dem der folgende Datenblock steht. Beim Directory-Sektor 18/01 ist dies normalerweise 18/04. Die Adresse für diesen nächsten Sektor steht immer in den ersten beiden Byte des aktuellen Sektors. Dies gilt für alle Arten von Dateien, auch für das Directory einer Diskette. Ein Programm tut also bei der Ausführung des Befehls »nächster Sektor« nichts anderes, als daß es die neue Tracknummer gleich dem ersten und die neue Sektornummer gleich dem zweiten Byte des Blocks setzt und dann diesen lädt.

Nun wollen wir uns aber langsam an die handfesten Beispiele und Tricks wagen. Dazu benötigen Sie eine Diskette zum Ausprobieren, auf der keine wichtigen Daten gespeichert sind. Formatieren Sie diese mit dem Namen »DISKMON. TEST« und speichern Sie anschließend folgendes kleine Programm:

```
10 REM******************************
20 PRINT"HALLO - DIES IST EIN PROBETEXT"
30 END
```

Im Directory sollte nun dieser Eintrag erscheinen:

```
1  "PROBETEXT"       PRG
```

Die Anzahl der freien Blöcke wird mit 663 angegeben – und genau das soll nun anders werden.

Eingriff in die BAM

Ändern Sie mit Ihrem Diskmonitor oder, zur Not, auch mit »LITTLE ED« das Byte Nr. 76 auf Sektor 0/Track18 von 19 ($13) nach 20 ($14). Die Nummer eines Byte wird übrigens immer von Null an gezählt, so trägt also das erste Byte eines Sektors die Nummer 0. Beachten Sie dies bitte bei Angaben dieser Art. Wenn Sie nun noch einmal das Inhaltsverzeichnis ansehen, so sind nun angeblich wieder 664 Blocks frei! Hat die Floppy etwa plötzlich Speicherplatz gewonnen?

Sie hat natürlich nicht, das wäre ja zu schön, um wahr zu sein. Wir haben nur dem Computer vorgegaukelt, der von »PROBETEXT« belegte Sektor sei nun wieder frei – was er aber natürlich nicht ist. Und das erkennt die Floppy auch. Wenn wir Ihr nun nämlich den Befehl

```
OPEN 15,8,15 <RETURN>
PRINT 15,"VO" <RETURN>
CLOSE 15 <RETURN>
```

schicken, wird die Anzahl der freien Blöcke wieder richtig mit 663 angegeben. Warum?

Die Manipulation auf Track18/Sektor 0 betraf die BAM (Block Ability Map = Block-Belegungstabelle). Das erste Programm wird auf einer leeren Diskette immer ab Track 19/Sektor 0 gespeichert. Byte 76 der BAM enthält die Anzahl der unbelegten Sektoren auf Track 19, und diese haben wir einfach auf den ursprünglichen Wert 20 geändert. Sie könnten aber trotzdem beliebig viele weitere Programme speichern, der Sektor mit Probetext würde nie überschrieben werden. Die Information, welche Sektoren im einzelnen nun belegt sind oder nicht, erscheint nämlich erst in den folgenden 3 Byte.

Um nun jedem Sektor die richtigen Bytes und Bits in der BAM zuordnen zu können, benützen Sie am besten das kleine Programm »BIT-SUCHER« (Listing 3). So sparen Sie sich einige Rechenarbeit. Beachten Sie aber, daß ein freier Sektor paradoxerweise mit einem gesetzten (!) Bit gekennzeichnet wird. Die BAM einer komplett gefüllten Diskette besteht also nur aus Nullen.

Kein Speichern mehr möglich

Um eine solche randvolle Diskette zu simulieren, müssen nur die Bytes 4 bis 143 mit Null beschrieben werden. Nun ist kein Speichern mehr möglich, bis der Validate-Befehl wie oben beschrieben durchgeführt wird.

Im ASCII-Dump des Sektors 0 auf Track 18 werden Sie sicher schon den Diskettennamen ab Byte 144 entdeckt haben. Diesen kann man ebenso wie die 5 Bytes ab Nr. 162 beliebig verändern. Die Bytes 162 und 163 enthalten übrigens das ID, 165 und 166 die Kennung »2A« für die DOS-Version beziehungsweise den Formattyp. Bye 164 enthält normalerweise ein geSHIFTetes Leerzeichen CHR$(160), dient aber nur zur Trennung.

Hier bieten sich zunächst die sogenannten illegalen Zeichen an, wie etwa Stern, Fragezeichen oder Anführungsstriche. Vom Basic aus führt die Verwendung solcher Zeichen in Disk- oder Dateinamen zu Fehlermeldungen, aber gegen den Diskmonitor kann sich die Diskette kaum wehren. Interessant ist aber auch die Wirkung von CHR$(147), also Bildschirm löschen. Das Verändern der Textfarbe ist übrigens genauso möglich wie die automatische Umschaltung auf den Textzeichensatz.

Viele C 64-Benutzer ärgern sich darüber, daß das Directory normalerweise als Programm geladen wird und somit andere Programme im Speicher löscht. Aber in bezug auf Diskettenschutz hat dies auch unbestreitbare Vorteile. Wenn Sie die Bytes 164-166 mit Nullen beschreiben, so wird künftig das Directory nur noch bis zum Disknamen ausgegeben und anschließend meldet sich der Computer wieder mit »READY«, denn drei Nullen sind die Markierung für das Ende eines Basic-Programms.

Jedoch gleich nach der Änderung ist von dem allem oft noch nicht viel zu sehen, das Inhaltsverzeichnis wirkt unverändert. Erst nach der Eingabe von

```
OPEN 15,8,15 <RETURN>
PRINT 15,"IO" <RETURN>
CLOSE 15
```

tritt der gewünschte Effekt ein. Dies hat seinen Grund darin, daß aus zeitlichen Gründen der Diskettenname mit ID nur einmal beim Diskettenwechsel in den Speicher der Floppy geladen und später nur von dort geholt wird, während der Diskmonitor wiederum nur auf die Diskette selber zugreift. Aber bei der Initialisierung mit »IO« wird der Name erneut von der Diskette geholt, nun sind auch alle Veränderungen wirksam.

Übrigens haben diese Manipulationen keinerlei Einfluß auf die Ladbarkeit der Programme, diese sind nach wie vor normal verfügbar, nur weiß eben niemand außer Ihnen, wie die Programme heißen. Unbefugte können nun maximal noch das erste Programm über LOAD "*",8 erreichen, es dürfte aber recht enttäuschend sein, wenn dieses erste

Programm ein sogenannter Dummy ist (das heißt der leere Basicspeicher wurde geSAVEt).

Es gibt auf Sektor 0 / Track 18 aber auch eine Möglichkeit, die Daten auf der Diskette auf andere Weise zu schützen, nämlich nur gegen Schreibzugriffe. Ändern Sie dazu Byte 2 in den Wert 64 statt normal 65 um.

Achtung! Diese Änderung kann nicht jeder Diskmonitor vornehmen. Sollte er zu stark Basic-orientiert sein, so fängt das Laufwerk diese »unerlaubte« Editierung ab und meldet einen »73, CBM DOS Vx.x 15xx«-Fehler. In Byte 2 wird nämlich die DOS-Version beziehungsweise deren Format auf der Diskette festgehalten. Der Code 64 bedeutet dabei ein lese-, aber nicht schreibkompatibles Format. Mit dem Top-Flop kann dieses Byte übrigens nur im Burst-Modus, also zusammen mit einer 1570/71 editiert werden.

Disketten nicht mehr schreibkompatibel

Sollte Ihnen also die Änderung gelungen sein, so werden Sie feststellen, daß zwar alle Dateien weiterhin verfügbar sind, aber »SAVE« oder »SCRATCH« nur zur Ausgabe des oben beschriebenen DOS-Fehlers führen. Dieser Schutz ist übrigens der einzige, der auch für viele Diskmonitore wirksam ist und durch Abänderung nur eines einzigen Bytes erreicht wird.

Die meisten einfachen aber wirkungsvollen Veränderungen kann man wahrscheinlich im Directory ab Sektor 1/Track 18 vornehmen. Hier fallen vor allem die Dateinamen auf, die mit geSHIFTeten Leerzeichen aufgefüllt werden.

Der erste entsprechende Code 160 (hexadezimal: A0) innerhalb des Eintrags bedeutet dabei das Ende des tatsächlichen Namens und veranlaßt die Inhaltsverzeichnis-Routine, Anführungsstriche auszugeben. Das läßt sich beweisen, indem wir das »P« des Namens PROBETEXT, also Byte 5 mit dem Wert 80, in den Wert 160 umändern. Im Inhaltsverzeichnis erscheinen jetzt zwei Anführungsstriche hintereinander und dann »ROBETEXT« (das Programm ist nun übrigens außer mit dem Dateinamen »*« nicht mehr ladbar). Man kann also auch Texte nach den Anführungsstrichen ausgeben, was, richtig angewandt, oft sehr praktisch ist. Wenn ein Dateiname nicht mehr als zehn Zeichen lang ist, können Sie nämlich folgende Bytefolge anhängen:

```
160 44 56 44 49 58
```

oder im Zeichencode:

<geSHIFTetes Leerzeichen>,8,1:

Wollen Sie nun nach der Ausgabe des Inhaltsverzeichnisses ein Programm laden, so überschreiben Sie einfach die Angabe der Dateilänge mit »LOAD« und drücken <RETURN>. Das geht doch wirklich wesentlich schneller als die Eingabe des kompletten Ladebefehls samt Namen.

Schnellere Eingabe des Ladebefehls

Die Verwendung von illegalen Dateinamen hat seinen Zweck zwar bestimmt mehr als Zierde denn als Schutz, Anfänger bleiben jedoch trotzdem leicht daran hängen. Zu den illegalen Zeichen in Dateinamen gehören »*«, »?« und »$«, <DELETE> ist zwar erlaubt, aber dennoch etwas umständlich zu benützen. Obwohl der Diskmonitor hier problemlose Verwendung erlaubt, möchte ich doch auch noch die Basic-Variante kurz vorstellen. Wollen Sie zum Beispiel ein Programm unter dem Namen »*« speichern, so funktioniert dies mit folgendem Befehl:

```
SAVE ""+CHR$(42),8 (RETURN)
```

Aber der Umgang mit mehreren »CHR$« wird auf die Dauer doch sehr unübersichtlich. Mit dem Diskmonitor sind Ihrer Kreativität dagegen von technischer Seite her keinerlei Grenzen gesetzt. So wirkt ein Inhaltsverzeichnis, in dem jede Datei in einer anderen Farbe erscheint, wirklich sehr fröhlich und übersichtlich.

Aber ein mit Steuerzeichen editiertes Inhaltsverzeichnis stimmt nicht immer fröhlich. Setzen Sie doch mal vor den eigentlichen Dateinamen diese Bytefolge:

```
145 88 17 157
```

oder ausgeschrieben:

<CURSOR aufwärts> <X> <CURSOR abwärts> <CURSOR links>

Nun sollte der vorhergehende Dateiname im Inhaltsverzeichnis immer mit »X« beginnen. Was schätzen Sie, wie oft ein Unerfahrener versuchen wird, dieses Programm zu laden, und jedesmal nur einen »FILE NOT FOUND«-Fehler zurückbekommt, da das »X« gar nicht zum Namen gehört. Aber auch der tatsächlich veränderte Name dürfte ohne Monitor nur sehr schwer zu erraten sein.

Ähnliche Wirkung hat das Einbauen von CHR$(0) in den Dateinamen. Denn CHR$(0) bewirkt, kurz gesagt, überhaupt nichts – außer, daß er zur korrekten Angabe des Dateinamens unbedingt nötig ist.

Die letzte Art des Ladeschutzes gilt allgemein, sie ist also nicht von Basic aus zu umgehen, sondern muß auch wieder mit dem Diskmonitor aufgehoben werden. Es ist ganz einfach das Einsetzen gleicher Dateinamen. Mit »LOAD« kann dann nämlich nur noch die erste Datei dieses Namens erreicht werden, alle anderen sind unzugänglich.

Zum Schluß noch eine kleine Hilfe: Der erste Dateiname eines Directory-Sektors beginnt bei Byte 5, die anderen ab diesem 5. Byte alle 32 Bytes (ein kompletter Dateieintrag ist nämlich genau 32 Byte, inklusive 5 unbenützten Byte, lang.

Datei-Attribute editieren

Außer den Dateinamen stehen aber auch die Dateiattribute, also Informationen über Typ, Länge und Position der Datei im Directory zur Verfügung. Davon wollen wir zuerst den Dateityp näher begutachten. Seine Position ist 3 Byte vor Beginn des Namens, also immer das dritte Byte in einer Dump-Zeile. Überschreiben wir also gleich einmal Byte 2 (nicht vergessen: Numerierung ab 0) auf Sektor 1/Track 18 mit 0.

Plötzlich ist das Programm aus dem Inhaltsverzeichnis verschwunden. Für welchen Dateityp steht 0 also? Genaugenommen bedeutet 0: gelöscht, nicht geschützt, nicht geschlossen. Letzteres hat allerdings nicht viel zu bedeuten, es dient nur zur Differenzierung zu gelöschten Files, die trotzdem noch mit der Angabe »DEL« im Inhaltsverzeichnis zu sehen sind. Erhöhen wir die 0 zu einer 1, so ist im Inhaltsverzeichnis der Eintrag

```
1 "PROBETEXT"     *SEQ
```

zu finden. Vielleicht wissen Sie schon aus eigener Erfahrung, daß ein solches Sternchen keine Auszeichnung für besonders gelungene Programme ist, sondern eine nicht ordentlich geschlossene Datei anzeigt. Die 1 bedeutet also: sequentiell, nicht geschützt, nicht geschlossen. Wenn der Wert nun noch mal um 1 erhöht wird, erscheint

```
1 "PROBETEXT"     *PRG
```

Anscheinend ändert sich im Moment nur die Art des Programms, nicht aber alle weiteren Informationen. Für diese sind nämlich die oberen 2 Bit des Wertes zuständig. Durch Addition von 64 und/oder 128 müßten sie sich demnach ändern. Versuchen Sie es zunächst mit 64+2 (2 als Kennung für Programme). Der neue Eintrag sieht danach so aus:

```
1 "PROBETEXT"      *PRG<
```

Das »<«-Zeichen bedeutet, daß der Löschschutz aktiviert wurde. Sie können das sofort ausprobieren, indem Sie die Anweisung zum SCRATCHen der Datei geben:

```
OPEN 15 <RETURN>
PRINT 15,"SØ :PROBETEXT"
CLOSE 15
```

Einfacher SCRATCH-Schutz

Danach ist der Eintrag des Programms aber dennoch (hoffentlich) unverändert im Inhaltsverzeichnis zu finden. Allerdings bietet diese Methode nur Schutz vor versehentlichem Löschen, wer es mit Absicht darauf anlegt, die Datei zu verändern, gibt einfach einen SAVE-Befehl mit dem Klammeraffen zum Ersetzen der alten Datei ein, und der Löschschutz war einmal.

Die Bezeichnung nicht geschlossen (»*«) ändert zwar letztendlich nichts an der Ladbarkeit einer Datei, jedoch fühlen sich manche sicher dadurch irritiert. Um den ganz normalen Zustand einer Datei anzuzeigen, muß der Typ (0 bis 4) zu 128 addiert werden, womit das Bit 6 für den Löschschutz nicht gesetzt und das Bit 7 für die ordentliche Schließung der Datei gesetzt ist. Damit hätten wir also wieder den ursprünglichen Wert 130, und der Eintrag lautet wieder wie am Anfang:

```
1 "PROBETEXT"      PRG
```

Anhand von Tabelle 1 können Sie übrigens sehr einfach den Code Ihres gewünschten Programmtyps mit den entsprechenden Attributen errechnen. Zusätzlich sei noch erwähnt, daß Dateien der Arten DEL, SEQ, USR und REL ungeachtet ihres tatsächlichen Aufbaus nicht wie Programme über den Befehl »LOAD« geladen werden können, also wiederum eine Art Leseschutz besitzen. Umgangen kann dieser werden, wenn man den normalen LOAD-Befehl etwas modifiziert und – um etwa eine sequentielle Datei namens »ADRESSEN« zu laden – eingibt:

Ausrechnen des Programmtyp-Wertes

0	Basis
0	für Typ DEL
1	für Typ SEQ
2	für Typ PRG
4	für Typ USR
5	für Typ REL
64	für geschützte Datei
128	für ordentlich geschlossene Datei
	= gesamter Typen-Wert

Tabelle 1. Die Codes für die einzelnen Programmtypen sind leicht zu errechnen

```
LOAD "ADRESSEN,S",8
```

Dies funktioniert auch bei USR-Dateien.

Leseschutz im Dateitypen

Die beiden Bytes zwischen Dateityp und -name stellen die Adresse des ersten Datensektors der Datei dar. Auf sie muß sich auch der Benutzer eines Diskmonitors stützen, wenn er eine Datei auf der Diskette verfolgen will. Natürlich kann man durch ihre Abänderung ebenfalls interessante und mehr oder weniger gewünschte Effekte erzielen. Suchen Sie dazu einen bis jetzt noch unbenützten Sektor auf der Diskette, was ja im Moment noch nicht so schwer sein dürfte. Belegen Sie ihn dann, indem Sie das entsprechende Bit in der BAM löschen (Sie können dazu das Programm »BIT-SUCHER« sowie die Umrechnungsroutine des TEDMON in/von Binärcode zu Hilfe nehmen). Die einfachere Methode ist aber die in Basic:

```
OPEN 15,8,15 <RETURN>
PRINT 15,"B-A"; Ø ; Track; Sektor <RETURN>
CLOSE 15 <RETURN>
```

Die Rechenarbeit wird Ihnen durch den Block-Allocate-Befehl nun von der Floppy abgenommen.

Beschreiben Sie anschließend Byte 1 des Sektors mit 5 (Anzahl der gültigen Bytes auf dem letzten Sektor der Datei) sowie in die Bytes 2 und 3 die Anfangsadresse des Basic-Speichers Ihres Computers in High-/Lowbyte-Schreibweise. Anschließend schreiben Sie die Adresse des Sektors noch in den Dateieintrag, und alles, was nun noch geladen wird, ist ein leerer Speicher.

Wer es gern extrem mag, kann auch die Bytes 2 und 3 auf dem Wert Null lassen, dann wird nämlich beim Laden die Zeropage des Computers hoffnungslos mit Nullen überschrieben, was in den meisten Fällen zu einem Absturz des Systems führt.

Wie alles Gute im Leben hat aber auch diese Methode einen Haken, denn nach einem »VALIDATE«-Befehl werden die vom wirklichen Programm belegten Sektoren in der BAM freigegeben und eventuell beim nächsten Schreibzugriff gelöscht. Aber auch hier gibt es eine einfache Methode, um die Daten sicher zu machen.

Programm scheinbar nicht mehr vorhanden?

Dazu muß nur vor dem Dateieintrag des zu schützenden Programms die Adresse eines ersten Sektors zum Beispiel auf 0/255 gestellt werden. Wird nun ein »VALIDATE« an die Floppy gesendet, so bleibt diese an dem Eintrag mit einem »ILLEGAL TRACK OR SECTOR«-Fehler hängen. Einen ähnlichen Effekt erreicht man übrigens, wenn man einen ungültigen Code für den Dateityp angibt. Sie sehen also – auch die intelligenteste Floppy kann mit einem Diskmonitor überlistet werden.

Die auf den Dateinamen folgenden 7 Byte wollen wir nicht genauer behandeln. Nur so viel sei gesagt: Die ersten 3 Byte werden zur etwas schwer zu verstehenden Verwaltung relativer Dateien benutzt, die folgenden 4 sind für beliebige Zwecke frei.

Eine Idee wäre bei Anwendung der oben beschriebenen Manipulation der Sektoradresse die Notiz der richtigen Adresse in diesem Bereich. Das bekannte Betriebssystem GEOS gibt hier übrigens die Adresse des sogenannten Info-Blocks an, in dem nähere Informationen über die jeweilige Datei festgehalten werden.

In den letzten beiden Bytes eines Dateieintrags steht schließlich die Länge der Datei im Format Low-/Highbyte. Eine besonders sinnvolle Anwendung hierzu ist mir nicht bekannt, es sei denn, Sie wollen einem Bekannten (nach Möglichkeit ein Laie) weißmachen, Sie hätten ein Programm von 65535 Blöcken Länge geschrieben (dazu beide Bytes auf 255 setzen, entspricht übrigens dem Inhalt von fast 99 einseitig bespielten Disketten).

Zu diesem Abschnitt möchte ich Ihnen noch abschließend ein kleines Programm namens »DIR KOMPLETT« (Listing 4) vorstellen. Es listet alle Einträge im Directory (auch die mit Dateityp Ø) der Reihenfolge nach und durchnumeriert auf. Zusätzlich erfahren Sie noch, auf welchem Sektor des Directory die einzelnen Einträge zu finden sind.

Mit »DIR KOMPLETT« können Sie wesentlich gezielter auf die Suche nach bestimmten Einträgen gehen und auch

übersichtlich Aufschluß über bereits gelöschte Dateien bekommen. Das Ergebnis kann übrigens auf jedem Drucker ausgegeben werden (Bild 2).

Die direkte Dateimanipulation

Nun wollen wir langsam den Track 18 verlassen und uns der direkten Datei-Manipulation zuwenden. Die Adresse des ersten Blockes einer Datei kann man ja aus den beiden Bytes vor dem Dateinamen entnehmen. In diesem ersten Block steht als erstes die Adresse des jeweils nächsten, man nennt dies die »Sektorverkettung«. Ist der Sektor der letzte der Datei, so lautet die Sektorverkettung 0, gefolgt von der Anzahl der gültigen Bytes auf diesem Sektor (inklusive den ersten beiden).

Darauf folgt bei Programm-Dateien die Ladeadresse, während bei sequentiellen Dateien gleich die eigentlichen Daten beginnen. Relative Dateien wollen wir wieder außer acht lassen, da die Änderung ihres Aufbaus anhand eines einfachen Diskmonitors zu komplex ist, um im Rahmen dieses Artikels ausführlich erklärt zu werden.

Der Inhalt einer USER-Datei ist völlig frei vom Anwender (daher der Name) definierbar, deshalb können keinerlei Aussagen über den Aufbau gemacht werden, ihn müssen die findigen Monitor-Benutzer selber ausknobeln.

Zunächst bietet sich natürlich an, die Sektorverkettung zu ändern. Hierbei gilt grundsätzlich das gleiche wie bei der entsprechenden Änderung im Directory.

Eine sinnvolle Möglichkeit zum Eingriff in die Ladeadresse ist für Anwender gegeben, die Programme für den C 64 auf dem C 128 schreiben. Die im C 128-Modus gespeicherten Programme können nämlich vom C 64 mit »LOAD "DATEINAME",8,1« nicht korrekt in den Basic-Speicher geladen werden. Also muß die Ladeadresse auf den Beginn des Basic-Speichers des C 64, also auf 2049 ($0801 hexadezimal) gesetzt werden. Im Format Low-/Highbyte bedeutet dies dann 01/08. Jetzt ist auch relatives Laden in den C 64 möglich.

Auch die Angabe der gültigen Bytes auf dem letzten Sektor einer Programmdatei in Byte 1 kann sinnvoll genutzt werden. So ist es möglich, hinter der Programmende-Markierung eines Basic-Programms (drei Nullen) einen Informationstext anzufügen, der durch die Erhöhung von Byte 1 um die Zahl der angehängten Zeichen mit in den Speicher geladen wird, im Programm jedoch nirgends in Erscheinung tritt. Den Text kann man dann mit Hilfe eines Speichermonitors oder einer entsprechenden kleinen Ausgaberoutine auslesen. Der Trick funktioniert allerdings nur, wenn auf dem Sektor noch genügend ungenutzte Bytes hinter der Programmende-Marke zur Verfügung stehen.

Verändern von Basic-Programmen

Zunächst wollen wir uns mit dem Verändern von Basic-Programmen auf der Diskette beschäftigen. Wie so oft sticht da eine Möglichkeit ins Auge, mit der die eigenen Programme wieder geschützt werden können. Wenn Sie nämlich die ersten 3 Byte des Programms, also Byte 4 bis 6 des Sektors, mit Null beschreiben, so lädt der Computer zwar das komplette Programm, erreichbar über LIST oder RUN ist es allerdings nicht mehr. Erst wenn die 3 Byte wieder ihre Originalwerte bekommen, ist das Programm ganz normal verwendbar. Diese drei Originalwerte speichern Sie am besten nach dem eben besprochenen Prinzip hinter dem Programm.

Der wohl am häufigsten genutzte Eingriff in Basic-Programmen ist die Änderung von REM-Zeilen. Damit hat es nämlich folgendes auf sich: alle ASCII-Zeichen, die auf eine REM-Anweisung folgen, werden beim LISTen des Programms vom Computer direkt auf den Bildschirm geschickt. Dies ermöglicht den Einbau von Steuerzeichen jeder Art und somit eine äußerst kreative Gestaltung des Listings.

Um Ihnen davon einmal ein Beispiel zu geben, was hier so alles möglich ist, ändern Sie am besten den Sektor, auf dem unser kleines Programm Probetext gespeichert ist, folgendermaßen ab:

```
00 : 00 55 01 1C 25 1C 0A 00
08 : 8F 93 90 2A 11 2A 96 2A
10 : 97 2A 91 2A 11 2A 11 2A
18 : 91 2A 12 2A 92 2A 0D 2A
20 : 0E 2A ...
```

Sie sehen, der Effekt ist überaus interessant. Die verwendeten Steuerzeichen wollen wir nun der Reihe nach erläutern, sie müßten übrigens auf C 16, C 64 und C 128 gleichermaßen funktionieren.

Doch erst zum Grundsätzlichen. Wir haben mit der Änderung die ersten der 30 Sternchen in der REM-Zeile des Programms überschrieben. Dies ist natürlich kein Problem, solange die ganze Information nur aus Sternchen besteht, und dadurch in jedem Dump leicht auffindbar ist.

```
TRACK 18  / SEKTOR 1
 1 : EASY-DUMP
 2 : TOP-FLOP.SUB
 3 : EASY-DUMP
 4 : LITTLE ED
 5 : BIT-SUCHER
 6 : DIR KOMPLETT
 7 : BC-SUCHER
 8 : DATEISCHNUEFFLER
TRACK 18  / SEKTOR 4
 9 : TOP-FLOP.KORR
 10 : TOP-FLOP.V3
 11 :
 12 :
 13 :
 14 :
 15 :
 16 :
```

Bild 2. »DIR KOMPLETT« listet alle Einträge im Directory auf. Hier das Beispiel eines Ausdrucks.

Im abgeänderten Dump hätten wir da schon unsere Schwierigkeiten, wenn wir nicht wüßten, daß jenes $8F (dezimal 143) in Zeile $08 der Code des Basic-Befehls »REM« ist. Man nennt den Code in diesem Falle ein Token. Der Computer speichert die Basic-Schlüsselwörter nämlich nicht im Klartext, was einfach viel zu speicheraufwendig wäre, sondern als Abkürzungen, den besagten Token.

Die Token belegen die Codes ab 128, da diese Werte in keiner anderen Weise im Programmtext verwendet werden. Mit dieser Information ist es nun natürlich überhaupt keine Schwierigkeit mehr, die richtige Zeile zu finden. Beendet werden alle Zeilen übrigens mit einem Null-Byte, auch dieses ist leicht zu finden.

Die Token

In unserem speziellen Fall folgt auf das Token $8F schon das erste Steuerzeichen, nämlich $93/147 dez. Es ist für das Löschen des Bildschirms zuständig, und wird auch von der Taste <CLR> ausgegeben. Deshalb ist der Programmtext vorher, Zeilennummer und REM, nicht mehr beziehungsweise nur sehr kurz zu sehen, und die Ausgabe

des ersten Sternchens erfolgt in der linken oberen Ecke des Bildschirms.

Vor dem ersten Sternchen ($2A) steht allerdings noch ein weiteres Steuerzeichen. Der Wert $90 bewirkt die aktuelle Farbe Schwarz, in der das nun folgende Sternchen auf dem Bildschirm erscheint. Die Verwendung von Farbsteuerzeichen kann sehr gut dazu benützt werden, die Übersichtlichkeit von Programmen zu unterstützen.

Farbige Listings

Es ist nämlich auf diese Weise möglich, verschiedenen Programmabschnitten verschiedene Farben zu geben, beziehungsweise thematisch zusammengehörende Teile gleich einzufärben. Es gibt sogar einen »C«-Compiler für den C 128, der diese Option schon im normalen Editor bietet. Sie sollten aber unbedingt beachten, daß es wenig Sinn hat, die standardmäßige Hintergrundfarbe Ihres Computers zu verwenden, es sei denn, Sie wollten den letzten Teil eines Programms vor ahnungslosen Fremden verbergen. Bleibt dabei am Schluß die Hintergrundfarbe die aktuelle Zeichenfarbe, so wird auch die READY-Meldung des Systems nicht erkennbar und der Eindruck eines Absturzes erweckt.

Das nächste Sternchen auf dem Bildschirm erscheint eine Zeile tiefer, was durch das Cursor-Steuerzeichen $11 erreicht wurde. Bei Informationszeilen von entsprechender Länge sowie mit guter Planung ist es ohne weiteres möglich, einfache Blockgrafiken im Listing auszugeben. Um dies zu verdeutlichen, habe ich eine kurze Zeichenfolge entworfen, die ein großes Pluszeichen auf dem Bildschirm bewirkt. Sie kann in jeden Kommentar mit 59 oder mehr Zeichen eingefügt werden.

```
: 93 1D 1D 1D 75 69 11 9D
: 62 11 9D 6A 63 69 11 9D
: 6B 9D 9D 63 9D 9D 75 11
: 9D 62 11 9D 6B 9D 9D 6A
: 9D 91 62 9D 91 69 9D 9D
: 63 9D 9D 6A 9D 91 75 36
: 6B 91 90 62 0D 11 11 11
: 11 11 11 ...
```

Grafik im Listing

Das Ergebnis ist doch wirklich sehr überzeugend, oder? Am einfachsten ist es übrigens, wenn man die Zahlenfolgen in Basic entwirft und ausprobiert. Dabei hilft diese kleine Routine:

```
10 READ X:PRINT CHR$(X);:Z=Z+1:IF X<> 0 THEN 10
20 PRINT Z:END
30 DATA ...
```

Schreiben Sie dabei Ihre Zeichen- und Steuercodes in die DATA-Zeilen ab 30. Der letzte Wert muß immer Null sein, abschließend gibt die Routine die Anzahl der Werte inklusive der Null aus.

Ein anderer möglicher Zweck von Cursor-Steuerzeichen in dieser Verwendung könnte die nachträgliche Abänderung von vorhergehenden Zeilen des Listings sein. Soll beispielsweise ein Paßwort eingegeben und mit dem richtigen verglichen werden, so überschreibt eine folgende REM-Zeile am besten das richtige Paßwort, mit dem verglichen wird, mit irgendwelchen anderen Zeichen. Auf diese Weise braucht das Programm keinen Listschutz mehr. Auch schlaue Knacker, die sich die Paßwortabfrage im Listing ansehen, werden sich die Zähne ausbeißen, es sei denn, sie lassen es ausdrucken...

```
WAHRSCHEINLICHE BC-TEXTE:
TRACK 14  / SEKTOR 5   / BYTE NR. 88
TRACK 14  / SEKTOR 11  / BYTE NR. 70
TRACK 14  / SEKTOR 11  / BYTE NR. 96
TRACK 14  / SEKTOR 17  / BYTE NR. 56
TRACK 14  / SEKTOR 18  / BYTE NR. 42
```

Bild 3. Beispiel eines Ausdrucks mit »BC-SUCHER«

```
EINTRAG NR. 3

DATEINAME........... EASY-DUMP

DATEITYP............ PROGRAMM
                     CODE: + 130

LAENGE IN BLOECKEN.. 5

SEKTORVERKETTUNG.... 19 / 1
                     19 / 7
                     19 / 13
                     19 / 2
                     19 / 8
```

Bild 4. Beispiel eines Ausdrucks mit dem »DATEISCHNUEFFLER« (Listing 6)

Noch vier Steuerzeichen haben wir in unserer Programm-Manipulation vorhin verwendet: zuerst $12. Hiermit wird die inverse Darstellung eingeschaltet; besonders wichtige Kommentare lassen sich so auffällig hervorheben. $92 macht das Ganze genauso wieder rückgängig wie der folgende »Carriage Return« $0D. Der CR setzt zudem den Cursor an den Anfang der nächsten Zeile und schaltet eventuell den sogenannten »Quote Mode« ab, in dem Steuerzeichen als inverse Grafikzeichen dargestellt werden und der somit für unsere Zwecke nicht interessant ist.

Das letzte Steuerzeichen schließlich ist als einziges nachträglich noch auf alle bisher ausgegebenen Zeichen wirksam, $0E schaltet nämlich den Klein-/Großschriftmodus ein.

Grundsätzlich gilt alles bisher Gesagte auch für die Texte nach PRINT-Befehlen, nur mit dem Unterschied, daß man hier auf den Unterschied zwischen Text und Variablen achten muß, und daß die Steuerzeichen nun natürlich nur beim Programmablauf wirksam sind, im Listing aber als inverse Grafikzeichen erscheinen.

Nach der Erkennung von PRINT-Token ($99/+153) können Sie also wirklich aktiv in den Programmablauf eingreifen. Oft ist es sinnvoll, in komplexen fremden Programmen unwichtige Texte oder Leerzeilen durch eigene erklärende Kommentare zu ersetzen. Allerdings kann man auch mit weniger sinnvollen Einschüben doch recht interessante Effekte erzielen, wenn etwa statt dem langerwarteten Rechenergebnis nach Stunden nur die lakonische Frage »WARTEST DU AUF WAS BESTIMMTES ???« am Bildschirm erscheint. Allerdings sollten Sie Ihre computerbegeisterten Mitmenschen nicht allzu sehr foppen, das könnte auch Auswirkungen auf die zwischenmenschlichen Beziehungen haben...

Außerdem möchte ich Sie bitten, in fremden Dateien keine Copyright-Vermerke zu ändern, denn diese gehören keinesfalls unter die Sparte »unwichtige Texte«!

Ausgabetexte in Basic-Programmen sind relativ leicht auch für Anfänger zu editieren. Anders sieht es da schon mit Maschinenprogrammen aus. Texte sind dort mit einem

Disassembler nur sehr schwer von Tabellen oder normalem Programmcode zu unterscheiden. Da hilft wirklich nur noch der Griff zum Monitor, und da meist nur wenig Wissen über Umfang und Aufbau der fremden Programme vorhanden ist, empfiehlt sich der Diskmonitor. Hier fallen Texte im ASCII-Code sofort im ASCII-Dump auf. Ob Steuerzeichen auch funktionieren, hängt aber von der Art der Ausgabe-Routine ab, möglicherweise hilft nur systematisches Ausprobieren.

Ausgabetexte in Maschinenprogrammen

In Maschinenprogrammen ist allerdings auch eine Textcodierung Standard, die Basic-Programmierer nur rudimentär zu nutzen wissen. Es ist die Nutzung des Commodore-eigenen Bildschirmcodes, dessen Werte im Bildschirmspeicher für die richtige Ausgabe von Zeichen sorgen. Es ist für die sowieso stark speicherorientierten Maschinenprogramme also am leichtesten, Texte im Bildschirmcode festzuhalten und dann direkt in den Bildschirmspeicher zu kopieren.

Das einzige Problem an der Sache ist, daß nur wenige Diskmonitore über deine Textdecodierung dieses Bildschirmcodes verfügen (Top-Flop gehört übrigens zu den wenigen). Sie müssen die Texte also mühevoll im Hexdump ausfindig machen und auch dort editieren.

Wenigstens eine Hälfte dieser Arbeit nimmt Ihnen das Programm »BC-SUCHER« (Listing 5) ab. Es durchsucht eine beliebige Datei nach Anhäufungen von Werten zwischen 0 und 26 (in diesem Fall die normalen Buchstaben) und gibt diese vermuteten Positionen mit Track, Sektor und Byte-Nummer auf Bildschirm oder Drucker aus. Natürlich können es sich bei diesen Anhäufungen auch um Tabellen oder ähnliches handeln, das richtige herauszufinden ist dann Ihre Sache, aber eine hohe Wahrscheinlichkeit ist immerhin gegeben. Bild 3 zeigt den Ausdruck nach dem Durchsuchen einer zwei Blöcke langen »MASTERTEXT«-Datei, die nur aus Bildschirmcode besteht.

Nachträgliches Ändern von Textdateien

Damit wären wir auch gleich bei einer neuen Anwendung des Diskmonitors, dem nachträglichen Verändern von Textdateien, wenn das entprechende Textprogramm nicht mehr zur Verfügung steht. Dazu müssen Sie aber das Aufzeichnungsformat des Textprogramms kennen, vor allem bei sogenannten Kompaktfiles. Auch ist das Ganze natürlich höchst unkomfortabel, aber es soll ja nur als Notlösung gedacht sein.

Bestimmt sind Ihnen schon Maschinenprogramme aufgefallen, die ganz normal mit »RUN« gestartet werden können. Der Trick ist einfach: vor dem eigentlichen Maschinenprogramm steht eine Basic-Zeile, die den Startbefehl »SYS« enthält. Im ASCII-Dump des Monitors erkennen Sie ihn am Token $9E und der darauffolgenden explizit dargestellten Adresse. Selbige ist einfach zu verändern. So können sie dabei zum Beispiel auf eine eigene kleine Routine richten, es bietet sich hierzu beim C64 der Bereich von 49152 bis 53247 an. Dort muß sich dann natürlich schon beim Start des Programms ihre eigene Routine befinden, die mit einem Sprung zur Originaladresse beendet wird. Ist die Routine noch nicht dort, so springt der Computer »ins Leere« und stürzt unter Umständen ab – wieder eine Möglichkeit, Programme zu schützen.

So kommen wir nun langsam zum Ende unseres kleinen Ausflugs in die Welt des Programmschutzes und der Datei-Manipulation, der farbigen Listings und verschwundenen Programme. Natürlich ist es nicht möglich, im Rahmen eines solchen Exkurses alle Möglichkeiten aufzuzeigen, die der geschickte Umgang mit einem Diskmonitor bietet. Aber es wurden doch viele Tips und Anregungen gegeben, und schließlich wird das wertvollste Wissen immer noch durch eigenes Probieren und Erfahren gewonnen.

Zum Schluß möchte ich Ihnen noch ein Programm vorstellen, das Ihnen bei Ihrer künftigen Arbeit mit dem Diskmonitor sicher einige Arbeit abnehmen wird, es ist Listing 6, der »DATEISCHNUEFFLER«. Die Ausgabe des Ergebnisses erfolgt wieder wahlweise über Bildschirm oder Drucker. Den Namen der zu untersuchenden Datei können Sie beliebig abkürzen, die Verwendung von Sternchen oder Fragezeichen als Joker ist nicht nötig, beziehungsweise gar nicht erlaubt (der »BC-SUCHER« hat übrigens das gleiche Eingabeformat).

Das Programm gibt Ihnen dann die Nummer des Eintrags im Directory (ab 1 gezählt), den vollständigen Dateinamen, den ausführlichen Dateityp sowie die komplette Sektorverkettung aus.

Wie so ein Ausdruck dann letztendlich aussehen kann, zeigt Bild 4.

Wir wünschen Ihnen nun viel Spaß beim Experimentieren mit Ihrem neugewonnenen Wissen über Disketten, Monitore und deren vielfältige Möglichkeiten.

(Nikolaus Huber/sk)

```
10 REM ********************************   <141>
20 REM **    AUSGABE EINES SEKTORS     **  <106>
30 REM **   ALS HEX- UND ASCII-DUMP    **  <135>
40 REM ** AUF BILDSCHIRM ODER DRUCKER **   <188>
50 REM ********************************   <181>
60 :                                       <036>
70 :                                       <046>
80 :                                       <056>
90 :                                       <066>
100 PRINT"{CLR,DOWN}--- SEKTOR-DUMP ---"   <041>
110 PRINT"{2DOWN}AUSGABE AUF{SPACE,RVSON}B
    {RVOFF}ILDSCHIRM                       <118>
120 INPUT"{7SPACE}ODER{SPACE,RVSON}D{RVOFF
    }RUCKER{4SPACE}";AU$                   <161>
130 IF AU$="B"THEN AU=3:GOTO 160           <086>
140 IF AU$="D"THEN AU=4:GOTO 160           <163>
150 GOTO 110                               <094>
160 PRINT"{2DOWN}TRACK,SEKTOR ";           <010>
170 INPUT TR,SE                            <130>
180 PRINT"{2DOWN}BITTE DISKETTE EINLEGEN U
    ND TASTE..."                           <009>
190 GET G$:IF G$=""THEN 190                <180>
200 OPEN 15,8,15                           <102>
210 OPEN 2,8,2,"#"                         <219>
220 PRINT#15,"U1";2;0;TR;SE                <179>
230 INPUT#15,A,B$,C,D                      <093>
240 IF A>0 THEN PRINT"{2DOWN}ACHTUNG!!":PR
    INT B$                                 <009>
250 IF A>0 THEN GOTO 160                   <093>
260 IF AU=3 THEN PRINT"{CLR}"              <196>
270 OPEN 3,AU                              <102>
280 FOR L=0 TO 31                          <166>
290 DA=L*8:GOSUB 600                       <235>
300 PRINT#3,HE$": ";                       <078>
310 FOR M=0 TO 7                           <157>
320 GET#2,DA$                              <094>
330 DA(M)=ASC(DA$+CHR$(0))                 <040>
340 NEXT                                   <096>
350 FOR N=0 TO 7                           <205>
360 DA=DA(N):GOSUB 600                     <005>
370 PRINT#3,HE$;                           <216>
380 NEXT                                   <136>
390 PRINT#3,"{2SPACE}";                    <174>
400 FOR N=0 TO 7                           <255>
```

Listing 1. »EASY-DUMP« gibt einen Disketten-Sektor auf Drucker oder Bildschirm aus.

```
410 IF DA(N)>31 AND DA<96 THEN PRINT#3,CHR
    $(DA(N));:GOTO 430                        <150>
420 PRINT#3,".";                              <126>
430 NEXT                                      <186>
440 PRINT#3                                   <181>
450 IF L<>15 THEN 500                         <253>
460 IF AU=4 THEN 500                          <230>
470 PRINT"{DOWN}TASTE..."                     <016>
480 GET G$:IF G$=""THEN 480                   <248>
490 PRINT"{CLR}"                              <224>
500 NEXT L                                    <098>
510 CLOSE 15:CLOSE 2:CLOSE 3                  <045>
520 END                                       <014>
600 D1=INT(DA/16)                             <166>
610 D2=DA-(D1*16)                             <220>
620 IF D1>=0 AND D1<=9 THEN D1$=STR$(D1)      <021>
630 IF D2>=0 AND D2<=9 THEN D2$=STR$(D2)      <116>
640 IF D1>9 THEN D1$=CHR$(55+D1)              <176>
650 IF D2>9 THEN D2$=CHR$(55+D2)              <208>
660 HE$=RIGHT$(D1$,1)+RIGHT$(D2$,1)+" "       <228>
670 RETURN                                    <220>
```

Listing 1. »EASY-DUMP« (Schluß)

```
10 REM ********************************       <141>
20 REM *** EDITIEREN EINZELNER BYTES ***      <070>
30 REM *** EINES BELIEBIGEN SEKTORS  ***      <054>
40 REM ********************************       <171>
100 PRINT"{CLR,DOWN}--- BYTE-EDIT ---         <027>
110 PRINT"{3DOWN}EINGABE TRACK,SEKTOR ";      <150>
120 INPUT TR,SE                               <080>
130 INPUT"{DOWN}BYTE NR.  ";BN                <093>
140 OPEN 15,8,15                              <042>
150 OPEN 2,8,2,"#"                            <159>
160 PRINT#15,"U1";2;0;TR;SE                   <119>
170 GET#15,A,B$,C,D                           <024>
180 IF A>0 THEN PRINT"{2DOWN}ACHTUNG!!";B$
    :GOTO 110                                 <129>
190 PRINT#15,"B-P";2;BN                       <153>
200 GET#2,BY$                                 <111>
210 BY=ASC(BY$+CHR$(0))                       <194>
220 PRINT"{2DOWN}BYTE NR."BN" HAT DEN WERT
    "BY                                       <101>
230 INPUT"{DOWN}NEUER WERT ";NB               <248>
240 NB$=CHR$(NB)                              <078>
250 PRINT#15,"B-P";2;BN                       <213>
260 PRINT#2,NB$;                              <097>
270 PRINT#15,"U2";2;0;TR;SE                   <232>
280 CLOSE 2:CLOSE 15                          <202>
```

Listing 2. »LITTLE ED«, ein Behelfs-Monitor.

```
10 REM  ******************************        <064>
20 REM  *** ZUORDUNG EINES SEKTORS ***        <004>
30 REM  *** ZUM ENTSPRECHENDEN BIT ***        <129>
40 REM  ***  DER BELEGUNGSTABELLE  ***        <053>
50 REM  ******************************        <104>
90 PRINT"{CLR,DOWN}--- BAM-ZUORDNUNG ---"     <033>
100 PRINT"{DOWN}(DIE NUMMERIERUNGEN BASIER
    EN AUF NULL)                              <011>
110 PRINT"{3DOWN}EINGABE TRACK,SEKTOR ";      <150>
120 INPUT TR,SE                               <080>
130 BB=3+(TR-1)*3+TR:BF=BB                    <192>
140 IF SE>-1 AND SE<8 THEN BB=BB+1:BI=SE      <030>
150 IF SE>7 AND SE<16 THEN BB=BB+2:BI=SE-8    <071>
160 IF SE>15 THEN BB=BB+3:BI=SE-16            <093>
170 PRINT"{DOWN}BAM: BYTE NR."BB              <172>
180 PRINT"{DOWN,5SPACE}BIT{2SPACE}NR."BI      <008>
190 PRINT"{DOWN,5SPACE}ANZAHL DER FREIEN B
    LOCKS AUF TRACK"TR"                       <087>
200 PRINT"{DOWN,5SPACE}IN BYTE NR."BF         <142>
```

Listing 3. »BIT-SUCHER«

```
10 REM ********************************       <148>
20 REM ***    AUFLISTUNG ALLER     ****       <173>
30 REM *** EINTRAEGE IM DIRECTORY ****        <235>
40 REM ********************************       <178>
70 ZD=1                                       <034>
80 FOR L=1 TO 16:VA$=VA$+CHR$(160):NEXT       <053>
90 PRINT"{CLR,DOWN}--- KOPLETTES DIRECTORY
   ---"                                       <007>
100 PRINT"{DOWN}(ALLE NUMMERIERUNGEN BASIE
    REN AUF NULL)                             <141>
110 PRINT"{2DOWN}AUSGABE AUF{SPACE,RVSON}B
    {RVOFF}ILDSCHIRM"                         <152>
120 PRINT"{7SPACE}ODER{SPACE,RVSON}D{RVOFF
    }RUCKER{4SPACE}";                         <014>
130 INPUT AU$                                 <123>
140 IF AU$="B"THEN AU=3:GOTO 170              <100>
150 IF AU$="D"THEN AU=4:GOTO 170              <177>
160 GOTO 110                                  <104>
170 PRINT"{2DOWN}DISKETTE EINLEGEN UND TAS
    TE..."                                    <229>
180 GET G$:IF G$=""THEN 180                   <106>
190 TR=18:SE=1                                <077>
200 OPEN 15,8,15:OPEN 2,8,2,"#"               <099>
210 PRINT#15,"U1";2;0;TR;SE                   <169>
220 INPUT#15,A,B$,C,D                         <083>
230 IF A>0 THEN PRINT"{2DOWN}ACHTUNG!!":PR
    INT B$:GOTO 170                           <155>
240 IF AU=3 THEN PRINT"{CLR}"                 <174>
250 OPEN 3,AU                                 <080>
255 PRINT#3,"TRACK"TR"  / SEKTOR"SE           <081>
260 FOR L=0 TO 7                              <099>
270 PRINT#15,"B-P";2;L*32+5                   <135>
280 DA$=""                                    <102>
290 FOR M=1 TO 16                             <220>
300 GET#2,A$                                  <215>
310 DA$=DA$+A$                                <059>
320 IF DA$=VA$THEN 500                        <218>
330 NEXT                                      <086>
340 PRINT#3,ZD": "DA$                         <091>
350 ZD=ZD+1                                   <248>
360 NEXT                                      <116>
370 PRINT#15,"B-P";2;0                        <164>
380 GET#2,NT$:GET#2,NS$                       <193>
390 NT=ASC(NT$+CHR$(0)):NS=ASC(NS$+CHR$(0)
    )                                         <049>
400 IF NS>20 THEN 500                         <147>
410 TR=NT:SE=NS                               <243>
420 PRINT#15,"U1";2;0;TR;SE                   <125>
430 GOTO 255                                  <096>
500 CLOSE 15:CLOSE 2:CLOSE 3                  <035>
```

Listing 4. »DIR KOMPLETT«

```
10 REM ********************************       <060>
20 REM ** SUCHEN VON BILDSCHIRMCODE- **       <050>
30 REM **     TEXTEN IN PROGRAMMEN   **       <136>
40 REM ********************************       <090>
50 :                                          <026>
60 :                                          <036>
70 TR=18:SE=1                                 <213>
80 PRINT"{CLR,DOWN}--- BC-TEXTSUCHER ---"     <147>
90 PRINT"{3DOWN}AUSGABE AUF{SPACE,RVSON}B{
   RVOFF}ILDSCHIRM"                           <080>
100 PRINT"{7SPACE}ODER{SPACE,RVSON}D{RVOFF
    }RUCKER{4SPACE}";                         <250>
110 INPUT AU$                                 <103>
120 IF AU$="B"THEN AU=3:GOTO 150              <072>
130 IF AU$="D"THEN AU=4:GOTO 150              <149>
140 GOTO 90                                   <126>
150 INPUT"{2DOWN}NAME DER DATEI ";NA$         <169>
160 IF AU=3 THEN PRINT"{CLR}"                 <094>
170 OPEN 3,AU                                 <000>
180 PRINT#3,"WAHRSCHEINLICHE BC-TEXTE:"       <085>
190 OPEN 15,8,15:OPEN 2,8,2,"#"               <089>
200 PRINT#15,"U1";2;0;TR;SE                   <159>
210 INPUT#15,A,B$,C,D                         <073>
220 IF A>0 THEN 150                           <075>
230 FOR L=0 TO 7                              <067>
240 PRINT#15,"B-P";2;L*32+5                   <103>
250 DA$=""                                    <070>
260 FOR N=1 TO LEN(NA$)                       <085>
270 NT=ET:NS=ES                               <013>
280 GET#2,A$                                  <195>
290 DA$=DA$+A$                                <039>
300 NEXT N                                    <170>
310 IF DA$=NA$THEN 400                        <196>
320 NEXT L                                    <174>
330 PRINT#15,"B-P";2;0                        <124>
340 GET#2,NT$:GET#2,NS$                       <153>
350 NT=ASC(NT$+CHR$(0)):NS=ASC(NS$+CHR$(0)
    )                                         <009>
```

```
360 IF NS>20 THEN 700                              <111>
370 PRINT#15,"U1";2;0;NT;NS                        <166>
380 GOTO 230                                       <110>
400 PRINT#15,"B-P";2;L*32+3                        <249>
410 GET#2,ET$:GET#2,ES$                            <091>
420 ET=ASC(ET$+CHR$(0)):ES=ASC(ES$+CHR$(0)
    )                                              <082>
430 PRINT#15,"U1";2;0;ET;ES                        <057>
440 NT=ET:NS=ES                                    <183>
450 FOR L=0 TO 255                                 <062>
460 GET#2,A$                                       <119>
470 A=ASC(A$+CHR$(0))                              <086>
480 IF A<27 THEN FT=FT+1                           <011>
490 IF A>26 THEN FT=0                              <030>
500 IF FT=5 THEN GOSUB 600                         <075>
510 NEXT L                                         <108>
520 PRINT#15,"B-P";2;0                             <060>
530 GET#2,NT$:GET#2,NS$                            <089>
540 NT=ASC(NT$+CHR$(0)):NS=ASC(NS$+CHR$(0)
    )                                              <201>
550 IF NS>20 THEN 800                              <049>
560 PRINT#15,"U1";2;0;NT;NS                        <102>
570 GOTO 450                                       <094>
600 PRINT#3,"TRACK"NT" / SEKTOR"NS" / BYTE
    NR."L-5                                        <186>
610 FOR M=L TO 255                                 <107>
620 GET#2,A$                                       <025>
630 A=ASC(A$+CHR$(0))                              <248>
640 IF A>63 THEN RETURN                            <202>
650 NEXT M                                         <002>
660 GOTO 520                                       <144>
700 PRINT"{DOWN}DATEI NICHT GEFUNDEN !!"           <165>
710 CLOSE 15:CLOSE 2:CLOSE 3                       <247>
800 PRINT"ENDE !!"                                 <154>
810 CLOSE 15:CLOSE 2:CLOSE 3                       <093>
```

Listing 5. »BC-SUCHER« sucht noch Bildschirmcodes in Dateien.

```
10 REM ********************************            <148>
20 REM * ANALYSIEREN EINES PROGRAMMS *             <140>
30 REM * NACH PROGRAMMTYP,ATTRIBUTEN *             <077>
40 REM *       SEKTORVERKETTUNG       *            <034>
50 REM ********************************            <188>
60 :                                               <036>
70 :                                               <046>
80 :                                               <056>
90 DIM DI(35):ZD=1                                 <014>
100 PRINT"{CLR}--- DATEI-ANALYSE ---               <061>
110 PRINT"{2DOWN}AUSGABE AUF{SPACE,RVSON}B
    {RVOFF}ILDSCHIRM"                              <152>
120 PRINT"{7SPACE}ODER{SPACE,RVSON}D{RVOFF
    }RUCKER{4SPACE}";                              <014>
130 INPUT AU$                                      <123>
140 IF AU$="B"THEN AU=3:GOTO 170                   <100>
150 IF AU$="D"THEN AU=4:GOTO 170                   <177>
160 GOTO 110                                       <104>
170 INPUT"{2DOWN}DATEINAME ";DN$                   <102>
180 PRINT                                          <026>
190 TR=18:SE=1                                     <077>
200 OPEN 15,8,15:OPEN 2,8,2,"#"                    <099>
210 PRINT#15,"U1";2;0;TR;SE                        <169>
220 FOR L=0 TO 7                                   <057>
230 PRINT#15,"B-P";2;L*32+5                        <093>
240 DA$=""                                         <060>
250 FOR M=1 TO LEN(DN$)                            <097>
260 GET#2,DX$                                      <174>
270 DA$=DA$+DX$                                    <002>
280 NEXT M                                         <142>
290 IF DA$=DN$THEN 400                             <184>
300 PRINT"#"ZD":"DA$                               <171>
310 ZD=ZD+1                                        <208>
320 DA$=""                                         <142>
330 NEXT L                                         <184>
340 PRINT#15,"B-P";2;0                             <134>
350 GET#2,TR$:GET#2,SE$                            <094>
360 TR=ASC(TR$+CHR$(0)):SE=ASC(SE$+CHR$(0)
    )                                              <040>
370 IF TR<1 OR TR>70 OR SE<0 OR SE>20 THEN
     1000                                          <115>
380 GOTO 210                                       <078>
400 PRINT#15,"B-P";2;L*32+2                        <241>
410 FOR N=0 TO 29                                  <192>
420 GET#2,DX$                                      <078>
430 DI(N)=ASC(DX$+CHR$(0))                         <164>
440 NEXT                                           <196>
450 OPEN 3,AU:PRINT#3                              <053>
460 PRINT#3,"EINTRAG NR."ZD                        <220>
470 PRINT#3                                        <211>
480 :                                              <202>
490 PRINT#3,"DATEINAME........... ";               <031>
500 DN$=""                                         <170>
510 FOR N=3 TO 18                                  <002>
520 DN$=DN$+RIGHT$(CHR$(DI(N)),1)                  <089>
530 NEXT                                           <032>
540 PRINT#3,DN$                                    <010>
550 :                                              <018>
560 PRINT#3:PRINT#3,"DATEITYP............
    ";                                             <027>
570 PRINT#15,"B-P";2;L*32+2                        <157>
580 GET#2,DX$                                      <240>
590 DT=ASC(DX$+CHR$(0)):DD=DT                      <088>
600 IF DT>=0 AND DT<=4 THEN FO=1                   <069>
610 IF DT>=128 AND DT<=132 THEN DT=DT-128          <185>
620 IF DT>=80 AND DT<=84 THEN FO=1:FG=1:DT
    =DT-80                                         <096>
630 IF DT>=192 AND DT<=196 THEN FG=1:DT=DT
    -192                                           <089>
640 IF DT=0 THEN PRINT#3,"GELOESCHT":GOTO
    700                                            <079>
650 IF DT=1 THEN PRINT#3,"SEQUENTIELL":GOT
    O 700                                          <092>
660 IF DT=2 THEN PRINT#3,"PROGRAMM":GOTO 7
    00                                             <207>
670 IF DT=3 THEN PRINT#3,"USER":GOTO 700           <162>
680 IF DT=4 THEN PRINT#3,"RELATIV":GOTO 70
    0                                              <239>
690 PRINT#3,"???                                   <100>
700 IF FO=1 THEN PRINT#3,"{21SPACE}NICHT G
    ESCHLOSSEN"                                    <115>
710 IF FG=1 THEN PRINT#3,"{21SPACE}GESCHUE
    TZT                                            <059>
720 PRINT#3,"{21SPACE}CODE: +";DD                  <023>
730 PRINT#3                                        <217>
740 :                                              <208>
750 PRINT#3,"LAENGE IN BLOECKEN..";                <120>
760 PRINT#15,"B-P";2;L*32+2+28                     <089>
770 GET#2,A$:GET#2,B$                              <094>
780 LL=ASC(A$+CHR$(0)):LH=ASC(B$+CHR$(0))          <112>
790 LD=LL+256*LH                                   <049>
800 PRINT#3,LD                                     <173>
810 PRINT#3                                        <043>
820 :                                              <034>
830 PRINT#3,"SEKTORVERKETTUNG....";                <004>
840 PRINT#15,"U1";2;0;TR;SE                        <037>
850 PRINT#15,"B-P";2;L*32+3                        <191>
860 GET#2,ET$:GET#2,ES$                            <033>
870 ET=ASC(ET$+CHR$(0)):ES=ASC(ES$+CHR$(0)
    )                                              <024>
880 PRINT#3,ET"/"ES                                <035>
890 TR=ET:SE=ES                                    <110>
900 PRINT#15,"U1";2;0;TR;SE                        <097>
910 PRINT#15,"B-P";2;0                             <196>
920 GET#2,NT$:GET#2,NS$                            <225>
930 NT=ASC(NT$+CHR$(0)):NS=ASC(NS$+CHR$(0)
    )                                              <081>
940 IF NT<1 OR NT>70 OR NS<0 OR NS>20 THEN
     1020                                          <046>
950 PRINT#3,"{20SPACE}";NT"/"NS                    <152>
960 TR=NT:SE=NS                                    <029>
970 GOTO 900                                       <200>
1000 PRINT"{2DOWN}FILE NICHT GEFUNDEN !!"          <039>
1010 CLOSE 15:CLOSE 2                              <222>
1020 CLOSE 15:CLOSE 2:CLOSE 3                      <047>
```

Listing 6. »DATEISCHNUEFFLER« gibt Ihnen wichtige Informationen.

So geht's mit Geos — das neue Floppy-Feeling

Die grafische Benutzeroberfläche »Geos« stellt ein eigenes Betriebssystem dar. Dadurch verändern sich alle Aspekte der C64-Bedienung; besonders wird die Arbeit mit Diskettenlaufwerken revolutioniert. Welche Neuerungen bringt Geos für den Floppy-Betrieb.

Wie vieles unter Geos ganz anders abläuft als früher, so ergibt sich auch eine völlig neue Disketten- und Laufwerksorganisation. Hier erfahren Sie Wege, um die durch Geos entstehenden Vorteile zu nutzen, und erhalten Hilfestellungen, um bestimmte Schwierigkeiten zu meistern. Dabei werden wir uns bis zu den Software-Grundlagen vorarbeiten. Sie sollten dazu Geos (am besten

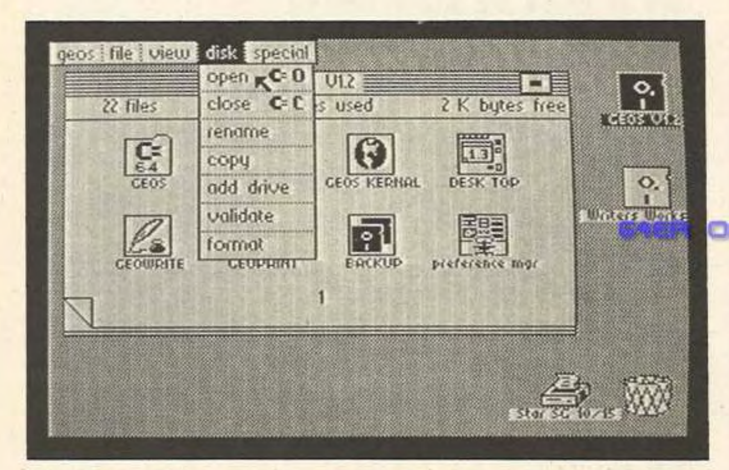


Bild 1. Das Menü »Diskette« des Geos-DeskTop bietet komfortable Diskettenoperationen

V1.3) oder Geos 128 besitzen, und bereits über ein gewisses Geos-Grundwissen verfügen; andernfalls können Sie diesen Artikel als »Erfahrungsbericht« zur sachlichen Information über Geos lesen.

Die Elemente der eigentlichen Benutzeroberfläche sind jedoch nicht Thema dieses Artikels (hier sei auf den Geos-Einsteigerkurs im Stammheft des 64'er-Magazins verwiesen), sondern nur die Floppy-Behandlung von Geos.

Floppy-Beschleuniger inbegriffen

Die starke Diskettenbezogenheit von Geos zeigt sich schon darin, daß Geos von Diskette geladen (gebootet) wird. Nach

```
LOAD"*",8,1
```

erscheint die Meldung "GEOS BOOTET ..."; nun wird Geos eingelesen. Dies geschieht bereits mit dem in Geos integrierten Floppy-Speeder »diskTurbo«, welcher softwaremäßig den Floppy-Zugriff um den Faktor 5 bis 6 beschleunigt. Eventuell vorhandene Floppy-Speeder schaltet »diskTurbo« einfach ab.

Auffällig an der Arbeitsweise von »diskTurbo« ist das hektische Flackern der LED (rote Lampe am Laufwerk) sowie das leise Schrittmotorgeräusch.

Wer ältere Geos-Versionen (bis 1.2) besitzt, kennt noch einen Nachteil des alten Geos-Kopierschutzes: Regelmäßige Abstürze beim Booten von Geos V1.2 waren keine Seltenheit – es konnte sogar vorkommen, daß selbst dreimal hintereinander »nichts ging«.

Mit Geos V1.3 für den C64 und Geos 128 gibt es diese Probleme überhaupt nicht mehr, obwohl das Programm immer noch gegen unbefugtes Kopieren geschützt ist.

Sobald man eine Diskette unter Geos »öffnet«, das heißt den OPEN-Befehl anklickt, wird diese auf ein Kriterium hin untersucht: Handelt es sich um eine unter Geos formatierte Diskette?

Falls nein, so wird sie wahlweise ins Geos-Format konvertiert; dieser Arbeitsschritt kann auf jede »normale« Programmdiskette angewandt werden, bei kopiergeschützter Originalsoftware sollte man jedoch lieber »NEIN« anklicken.

Geos erkennt »seine« Disketten an einer besonderen Markierung im Directory, welche im normalen Directory jedoch nicht sichtbar ist.

An dieser Stelle ist eine wichtige Warnung angebracht:

> Bearbeiten Sie Geos-formatierte oder ins Geos-Format konvertierte Disketten niemals außerhalb von Geos. Ansonsten riskieren Sie Datenverluste;

Dieser Hinweis bezieht sich nicht auf reine Lesezugriffe (Laden bzw. Lesen von Dateien), die den Inhalt einer Diskette nicht beeinflussen; das Schreiben von Programmen oder Daten sowie deren Veränderung hingegen sind für Geos-Disketten »lebensgefährlich«.

Eine Ausnahme bilden selbstverständlich Hilfsprogramme, die speziell für Geos entwickelt wurden, aber nicht unter Geos ablaufen. Andere Utilities hingegen sollten Sie nur auf gewöhnliche C64-Disketten anwenden.

Warum dies so ist, erfahren Sie an späterer Stelle in diesem Artikel, wenn wir auf die programmtechnischen Grundlagen zu sprechen kommen.

Die Diskettentypen

Geos unterscheidet insgesamt zwischen drei verschiedenen Sorten von Disketten, wovon man als Anwender nur die erste mittels Formatierung erstellen kann:

1) Arbeitsdisketten

Auf diese Disketten lassen sich alle verfügbaren Operationen anwenden. Solche Disketten entstehen automatisch, wenn C64-Disketten ins Geos-Format konvertiert oder neue Disketten unter Geos formatiert werden.

2) Systemdisketten

Solche Disketten enthalten ein bootfähiges Geos-System und genießen besonderen Schutz vor versehentlichem Überschreiben oder Kopieren.

3) Hauptdisketten

Zu Geos sind zahlreiche Applikationen erhältlich; deren Originaldisketten werden als »Hauptdisketten« besonders gekennzeichnet und ähnlich wie Systemdisketten behan-

delt. Damit soll insgesamt sichergestellt sein, daß Sie keinen Schaden auf teuer erworbenen Programmdisketten anrichten.

Unter Geos darf man niemals ohne weitere Operationen eine Diskette wechseln, sondern muß nach Einlegen der neuen Diskette diese »öffnen«, was unter DeskTop durch Anklicken des Diskettensymbols geschieht.

Der umgekehrte Vorgang des Schließens ist theoretisch ebenfalls vonnöten, kann in der Praxis jedoch entfallen.

Dennoch stellt die »Öffnungspflicht« eine gewaltige Umstellung für alle Anwender dar, die aber zu bewußterem Umgang mit Disketten erzieht.

Dies heißt jedoch nicht, daß Geos keinen unerlaubten Diskettenwechsel bemerkt: Anhand der Diskettennamen unterscheidet Geos jede Diskette von der anderen. Geben Sie deshalb bei der Formatierung von Geos-Disketten jeder Diskette einen eigenen Namen; dann kann bei der Anwendung nichts passieren. Zudem sollten Sie jede Diskette mit ihrem Diskettennamen beschriften, was sich in der Praxis sehr bewährt.

Das Menü »Diskette«

Der DeskTop bietet verschiedene Möglichkeiten zur Behandlung von Disketten in einem eigenen Pull-Down-Menü: »Öffnen«, »Schließen«, »Umbenennen«, »Kopieren«, »Laufwerk anmelden«, »Aufräumen (Validieren)« und »Kopieren«. Bild 1 zeigt dieses Menü.

Die Bezeichnungen lassen schon erahnen, welche Möglichkeiten sich hinter den einzelnen Menüpunkten verbergen. »Öffnen« kann durch die Tastenkombination <CBM O>, »Schließen« durch <CBM C> ersetzt werden; dies kommt einem schnelleren Arbeiten sehr zugute.

Das Umbenennen einer Diskette entspricht dem Floppy-Befehl »R« (Rename); das Kopieren ist für Besitzer von Systemen mit zwei Laufwerken von größtem Nutzen, doch mit

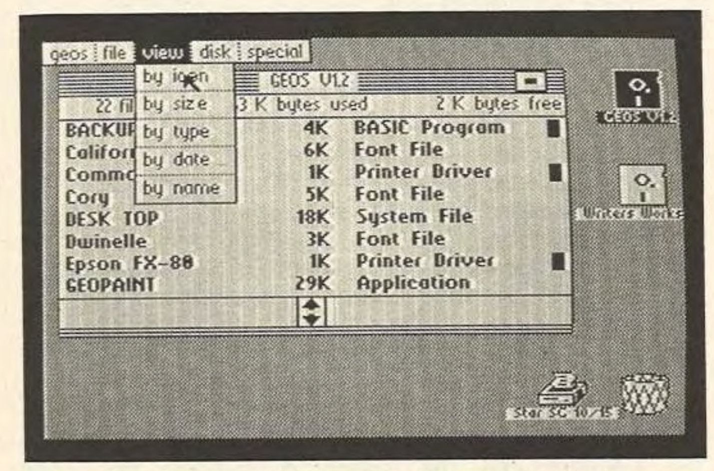


Bild 2. Anzeige des Disketteninhalts in alhabetischer Reihenfolge (oben ist auch das Menü »Anzeige« zu sehen)

nur einem Laufwerk sind 30fache (!) Diskettenwechsel keine Seltenheit.

Verwenden Sie dann lieber das mitgelieferte Backup-Programm »DISKETTENKOPIER«.

Das »Aufräumen« ist schließlich der letzte Rettungsanker, wenn man außerhalb von Geos einen Befehl wie V (Validate) oder S (Scratch) versehentlich auf eine Geos-Diskette angewendet hat; solange keine Daten mehr auf die dadurch beschädigte Diskette gespeichert werden, hilft die Auswahl von »Aufräumen« im Menü »Diskette« auf jeden Fall weiter.

Nie mehr LOAD "$",8

Die Auflage, zur Ansicht eines Directories im C 64-Betrieb die Befehle LOAD "$",8 und LIST einzugeben, ist eine Zumutung für jeden Anwender. Geos zeigt unter DeskTop au-



Bild 3. Ein Info-Bildschirm der Option »Datei/Info« gibt Auskunft über das gerade angeklickte Piktogramm

tomatisch den Inhalt der aktuellen Diskette an, wie Sie auch in Bild 1 sehen konnten: Jedem Programm ist neben dem Filenamen auch eine Kleingrafik (Icon, Piktogramm) zugeordnet; pro Bildschirmseite werden acht Dateien dargestellt. Durch Umschalten mit den Zifferntasten oder dem »Eselsohr«-Symbol blättert man bequem vor oder zurück.

Der Vorteil der grafischen Darstellung liegt vor allem in der erhöhten Anschaulichkeit, die gerade Einsteigern den Zugang zur Computerwelt erleichtert. So orientieren sich alle Elemente der Dateibehandlung immer an diesen Kleingrafiken, welche meistens versuchen, die Funktion einer Datei zu symbolisieren.

Die Anzeige des Inhaltsverzeichnisses läßt sich jedoch auch umschalten: Wählen Sie dazu im Menü »Anzeige« den entsprechenden Punkt an. Bild 2 zeigt die Anzeige der Dateien »nach Namen«.

Dateibehandlung unter Geos

Nun werden wir uns ausführlicher damit auseinandersetzen, wie Geos die Dateien einer Diskette behandelt.

Zunächst wollen wir die elementaren Operationen wiederholen:

- Durch »Doppelklicken« einer Datei wird sie geöffnet, was bei Programmdateien dem Start entspricht. Ein besonderer Service wird deutlich, wenn man Text- oder Grafikdateien anklickt: Geos lädt, sofern möglich, das dazugehörige Geos-Programm (GeoPaint oder GeoWrite) automatisch nach, und dieses gibt sofort die selektierte Datei zur Bearbeitung frei. Dies ist in der Praxis eine erhebliche Erleichterung; Geos versucht nämlich, jede Aktion des Anwenders einigermaßen sinnvoll zu interpretieren.
- Klickt man eine Datei einmal an, invertiert sich deren Piktogramm am Bildschirm. Nun sind die Operationen des »Datei«-Menüs auf diese ausgewählte Datei anwendbar.
- Bei »Pause-Doppelklick« wird eine Datei »beweglich«, sie kann nun am Bildschirm mit dem Joystick verschoben werden. Durch »Abklicken« auf dem Papierkorb wird die Datei gelöscht, das Abklicken auf dem Drucker startet einen Ausdruck der betreffenden Grafik- oder Textdatei. Die

Ablage auf dem »Rand« (unterer Bildschirmbereich) ermöglicht erweiterte Operationen wie das Kopieren von Dateien oder die Neustrukturierung des Disketteninhaltes (siehe 64'er 12/87, Geos-Einsteigerkurs).

Im folgenden wollen wir einige interessante Möglichkeiten der Dateibearbeitung herausgreifen.

Information ist alles

Geos speichert zu jeder Datei neben dem Filenamen eine Reihe weiterer Daten, welche auf einfache Weise an den Bildschirm zu holen sind: Klicken Sie eine Datei an (Mauszeiger darauf bewegen und den Joystick-Knopf drücken) und wählen Sie den Punkt »Info« im Menü »Datei«. Bild 3 zeigt ein Beispiel für diesen Informationsbildschirm.

Befindet sich auf einer Diskette die Datei »DESK TOP«, so kann der Informationstext — wie in Bild 3 zu sehen — editiert werden. Dadurch steht Ihnen die Möglichkeit offen, jede Datei mit einem kurzen Begleittext zu versehen, welcher unter DeskTop abrufbar ist. Bei sinnvollem Einsatz der Infotexte nimmt die Ordnung auf einer Diskette erheblich zu.

Die Datei-Spezifikationen

Folgende Informationen werden also zu einer Datei im Infobildschirm angezeigt:
- Dateiname
- Name der Diskette, von welcher die Datei stammt
- Dateityp
- Klasse
- Struktur
- Größe
- Datum und Zeit der letzten Änderung
- Autor
- Schreibschutz
- und der Infotext

Diese Begriffe stellen größtenteils eine Ergänzung derjenigen Informationen dar, welche die Floppy von sich aus vermerkt.

Der »Dateiname« ist eine bis zu 16 Zeichen lange Bezeichnung einer Datei, welche zu deren eindeutiger Bestimmung dient; außerhalb von Geos ist der Dateiname die einzige Möglichkeit für den Anwender, eine Datei zu finden, durch Geos werden noch weitere Unterteilungen getroffen.

```
    **** COMMODORE 64 BASIC V2 ****
 PROLOGIC SYSTEM 38911 BASIC BYTES FREE
READY.
RUN
      DISK BACKUP/RESTORE UTILITY
INSERT DESTINATION DISK TO BE FORMATTED
AND ENTER 'F' TO FORMAT, OR ENTER 'Q'
TO QUIT (F/Q)?
```

Bild 4. Aufforderung zum Einlegen einer Kopierdiskette

So differenziert Geos zwischen 15 verschiedenen Dateitypen, welche allesamt eine eigene Funktionsweise aufweisen und strikt voneinander getrennt werden:

»C= PRG«, »Basic-Programm«, »Assemblerprogramm«, »reine Daten«, »Systemdatei«, »Hilfsprogramm«, »Anwendung«, »Dokument«, »Schriftart«, »Druckertreiber«, »Disk-Treiber«, »Startprogramm«, »Temporär« und »selbstausführend«.

Weitere Werte kennt Geos nicht; der Dateityp »Disk-Treiber« wird außerdem von Geos 64 noch nicht genutzt.

Sehr interessant ist »Temporär«: Solche Dateien löscht Geos nach einmaliger Verwendung wieder, da sie nur ein einziges Mal zum Zwischenspeichern dienen. Deshalb wird man als reiner Anwender niemals die Gelegenheit haben, ein Temporär-File zu sehen.

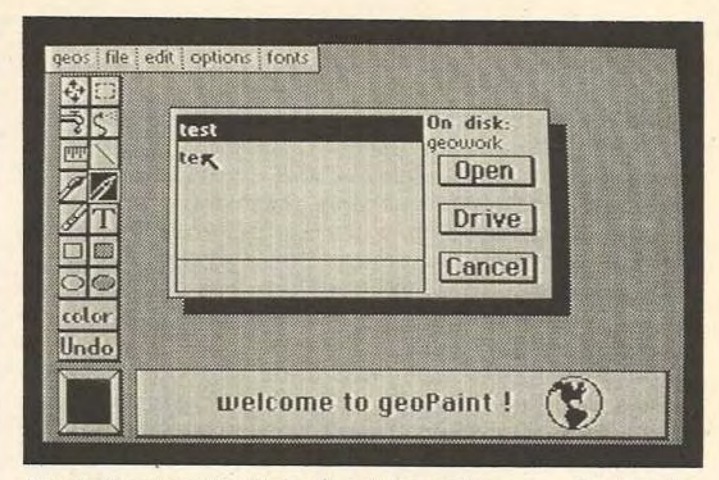


Bild 5. Die übersichtliche Datei-Auswahlbox von GeoPaint

Nun aber zu den zahlreichen anderen Angaben im Informationsbildschirm einer Datei.

Der im Info-Fenster angegebene »Disketten-Name« ist bei solchen Dateien wichtig, welche noch am Bildschirmrand zur Bearbeitung stehen, aber nicht von der aktuell eingelegten Diskette stammen. Vor allem beim Kopieren einer Datei kommt dem Disketten-Namen zentrale Funktion zu, denn Geos nennt beim Kopieren nicht »irgendeine« Quell- oder Zieldiskette, sondern die entsprechenden Disketten-Namen (Bild 4).

Von den Infofenster-Inhalten kennen Sie auch bereits die »Größe« einer Datei, aber unter Geos erfolgt diese in KByte; multiplizieren Sie diese Zahl mit 4, so erhalten Sie in etwa die Blockzahl. Da die Directories von Geos-Disketten auch im Normalbetrieb über LOAD "$",8 einlesbar sind, ist sogar die Ermittlung der exakten Blockzahl möglich.

Infrastruktur einer Datei

Der »Schreibschutz« macht sich bei LOAD"$",8 durch das Kleiner-Zeichen »<« hinter dem Dateityp bemerkbar; Geos erlaubt es, diesen durch das Anklicken des Schreibschutz-Quadrates jederzeit umzuschalten. Ohne Geos bräuchte man dazu ein spezielles Hilfsprogramm, welches nicht annähernd so komfortabel wie der DeskTop wäre.

Fassen wir nun zusammen: Völlig neu sind nur die Datei-Attribute »Klasse«, »Struktur«, »geändert«, »Autor« und der bereits erwähnte Infotext.

Der Zeitpunkt, welcher unter »geändert« erscheint, wird automatisch bei jedem Schreibzugriff auf eine Datei vermerkt; es handelt sich dabei um das laufende Datum und die aktuelle Zeit der Geos-Uhr, welche über den »Voreinstellungs-Manager« jederzeit einstellbar ist. Vor allem beim Erstellen von Programmen und Textdateien ist es für manche Anwender von Interesse, den Zeitpunkt der letzten Änderung zu wissen. Auf anderen Computern (PCs, Amiga) ist dies Standard.

Für kurze Kommentare bietet sich der Infotext an, welcher das untere Drittel im Fenster belegt. Dieser Text ist jedoch nur verfügbar, solange sich DeskTop auf der aktuellen Diskette befindet; zur Anzeige und Änderung des Infotextes muß Geos nämlich einen Teil der Datei »DESK TOP« nachladen. Dies ist ein reines Speicherplatz-Problem der C64-Version und tritt bei Geos 128 nicht mehr auf.

»Autor« bietet nun jedem stolzen Programmentwickler die Gelegenheit, sich hier zu verewigen.

Die »Struktur« einer Datei wird entweder als »SEQUENTIELL« (Format von C 64-Dateien wie PRG, SEQ und USR) oder »VLIR« (ein Geos-spezifisches Dateiformat) bezeichnet. Das VLIR-Format haben alle größeren Programme wie GeoWrite, GeoPaint oder der DeskTop, sowie alle Dokumente von GeoWrite, GeoPaint, GeoFile etc.

Wenn Sie sich gefragt haben, warum GeoWrite-Texte oder Notizblöcke erheblich länger werden können als sonstige C 64-Dateien (bis zu 127 Seiten werden erfaßt), so liegt dies im VLIR-Format begründet: Jede Textseite stellt gewissermaßen eine eigene Datei dar, aber die Dateien aller Textseiten behandelt Geos zusammen als eine einzige VLIR-Datei. Dieses flexible System des »Variable Length Indexed Record« erfreut auch Programmiererherzen, da nun Overlay-Software (Programme, die nach Bedarf »nachladen«) mühelos zu entwickeln ist.

Langer Rede kurzer Sinn: Die VLIR-Dateien sind eine geniale Mischung aus sequentiellen und relativen Dateien (DOS-Filetyp REL).

Nun zu einem weiteren Begriff des Infofensters.

Klassengesellschaft?

Bislang wurde die Funktion einer Datei bereits durch zwei extreme Differenzierungen bestimmt: Zum einen ist der Dateiname individuell wählbar und muß bei jeder Datei anders lauten, zum anderen gibt es 15 Dateitypen, von welchen jede Datei einem bestimmten Typ zugeordnet wird.

Einen Mittelweg schlägt die »Klasse« ein; dies ist eine bis zu 20 Zeichen lange Benennung, welche zwar ebenfalls bei jeder Datei anders heißen darf, aber in der Praxis einer bestimmten Gruppe von Dateien eigen ist. So gehören beispielsweise alle Dateien von »GeoPaint« der Klasse »Paint Image V1.0« an. Dies wiederum ermöglicht GeoPaint das zielsichere Auffinden aller seiner Bilddateien, obwohl sich diese vom Dateityp (jeweils »Dokument«) oder Dateinamen her nicht von GeoWrite-Texten abgrenzen ließen; Bild 5 zeigt dazu eine Datei-Auswahlbox von GeoPaint. Diese Form zählt zu den luxuriösesten Verfahren der Datei-Selektion, die auf Computern realisiert werden. Durch Anklicken eines Dateinamen wird dieser invertiert, und der jeweils hervorgehobene Name wird durch »OK« ausgewählt.

Vom Anwender ist die »Klasse« einer Datei nicht zu modifizieren; er ist jedoch durch die Option »Info« des Menüs »Datei« jederzeit in der Lage, dadurch eine nähere Vorstellung vom Wesen einer Datei zu beziehen.

Insgesamt also gibt Geos viele Informationen zu seinen Dateien bekannt, um die Anwendung zu erleichtern.

Neben den Applikationen (Anwendungen) und der Systemsoftware bietet Geos noch eine weitere Art von Programmen: die »Hilfsmittel«, auch »Schreibtischzubehör« genannt.

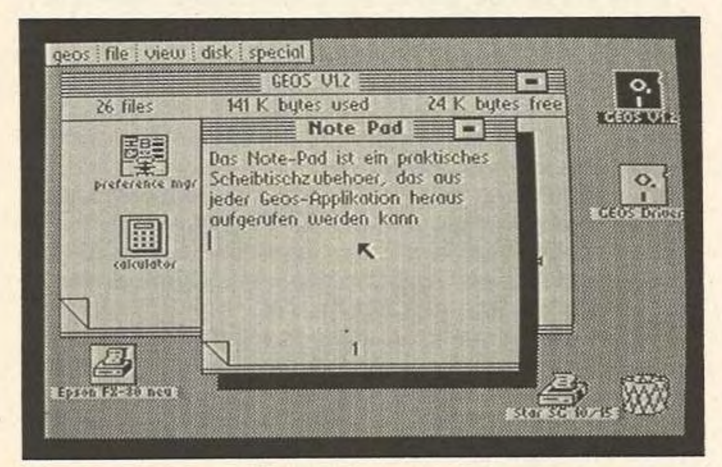


Bild 6. Notizblock, aus GeoWrite heraus aufgerufen

Dies sind Programme, die von anderen Programmen aus aufgerufen werden können, ohne daß dadurch das ursprüngliche Programm verlassen wird oder Daten verlorengehen. Bild 6 zeigt, wie aus der Textverarbeitung GeoWrite heraus der »Notizblock« eingelesen wurde; nach Verlassen des Notizblockes kann man mit GeoWrite fortfahren, als wäre der Notizblock nie geladen worden.

Folgende Hilfsmittel (außer dem Notizblock) enthält die Geos-Grundausstattung: »Wecker« zum Einstellen der Alarmzeit, »Voreinstellung« zum Verändern der Bildschirmfarben, des Mauszeigers oder der Mausgeschwindigkeit, »Rechner« als simulierter Taschenrechner sowie »Foto Manager« und »Text Manager« zum Verwalten wiederholt benötigter Text- und Grafikausschnitte.

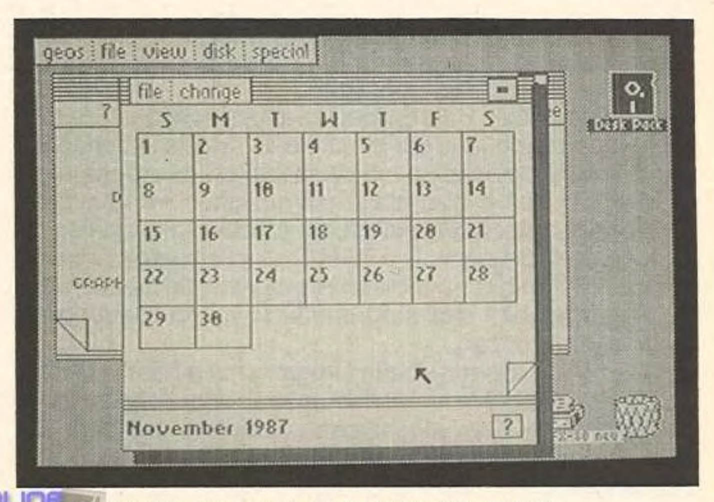


Bild 7. Terminkalender als Schreibtischzubehör

Weiteres Schreibtischzubehör wie ein optisch ansprechendes »Black jack« oder ein toller Terminkalender (Bild 7) sind separat im »Desk Pack 1« erhältlich.

Grafik- und Textmodule

Außerhalb von Geos gibt es Dutzende von Grafik- und Textformaten. Die wichtigsten davon sind mit Hilfe von Zusatzprogrammen wie »Graphics Grabber« und »Text Grabber« ins Geos-Format übertragbar.

Eine im C 64/128-Bereich einzigartige Besonderheit stellt noch das flexible Umgehen mit Grafikausschnitten dar. Diese lassen sich jederzeit aus GeoPaint-Bildern »herausschneiden« und in GeoPaint, GeoWrite oder der Datenbank »GeoFile« weiterverwenden. Verfügt man erst einmal über eine gewisse Grafikbibliothek, macht das Arbeiten mit Geos doppelt soviel Spaß, denn die Kooperation der einzelnen Geos-Programme ist nahezu optimal.

Bild 8 zeigt, wie problemlos Grafiken in Text eingebunden werden, und in Bild 9 findet sich eine Grafik in der Eingabemaske von GeoFile wieder.

Die Konfiguration

Eine große Neuerung von Geos V1.3 und Geos 128 besteht darin, daß der Betrieb der RAM-Erweiterungen 1700 und 1750 (Bild 10) kompromißlos unterstützt wird.

Obwohl diese Hardware laut Aufschrift nur für den C128 entwickelt wurde, läßt Geos deren Betrieb am C 64 uneingeschränkt zu.

In einem Konfigurationsmenü hat man die Möglichkeit, neben einem herkömmlichen Laufwerk auch eine sogenannte RAM-Disk zu betreiben.

Eine RAM-Disk ist eine weitere Diskettenstation mit der üblichen Kapazität; sie existiert jedoch nicht als reale Peripherie, sondern wird im Speicher der 512K-RAM-Erweiterung simuliert. Das bedeutet, Sie können in die RAM-Disk schreiben, Dateien aus ihr laden, Dateien in die RAM-Disk oder von ihr kopieren — wie wenn es sich um ein normales Laufwerk handeln würde. Der große Vorteil liegt jedoch darin, daß alle Operationen mit einer mindestens hundertfachen, oft noch viel höheren Geschwindigkeit ablaufen; verantwortlich dafür ist, daß die RAM-Disk über unmittelbare Speicherzugriffe funktioniert, was natürlich viel schneller sein muß als das Lesen von einer Diskette.

Beschattung gefällig?

Der elementare Unterschied zu einer richtigen Diskette liegt jedoch darin, daß der Inhalt der RAM-Disk ebenso wie der normale Speicher nach Ausschalten des Computers gelöscht ist; soll er also dauerhaft gesichert werden, ist er vor Beenden einer Arbeitssitzung auf eine »physikalische« Diskette zu übertragen. Die RAM-Disk ist also eine reine Arbeitsdiskette mit riesiger Geschwindigkeit (die meisten Diskettenoperationen sind aufgrund ihrer kurzen Dauer nicht mehr wahrnehmbar).

Die üblichen Schritte beim Umgang mit einer RAM-Disk als Zweitlauf bestehen also im Kopieren aller Arbeitsfiles in die RAM-Disk, dem eigentlichen Arbeiten und dem abschließenden Speichern des Ergebnisses auf eine 5¼-Zoll-Diskette.

Ohne jede Übertreibung wird Geos durch die RAM-Disk zu einem professionellen System, weil die vielen Diskettenzugriffe, die sonst leider den Komfort einschränken, bei einer RAM-Disk nicht mehr festzustellen sind. Vor allem dem

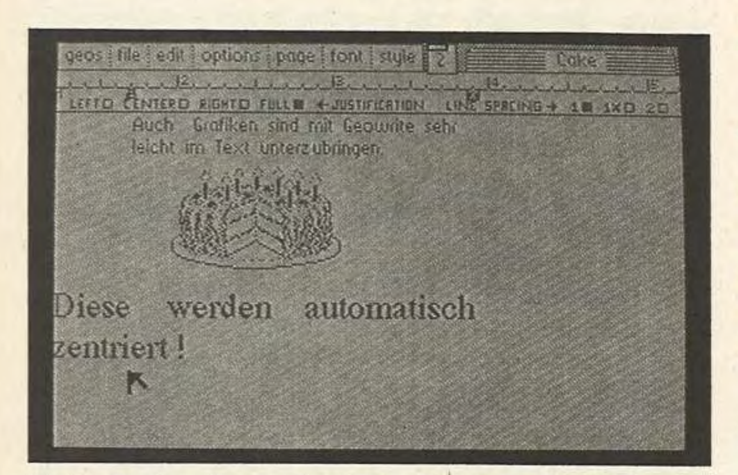


Bild 8. Grafiken lassen sich in GeoWrite-Texte einbinden

Arbeiten mit längeren Dateien sowie dem Kopieren kommt dies sehr zugute.

Wer jedenfalls einmal mit einer RAM-Disk gearbeitet hat, kann sich nicht mehr vorstellen, wie er ohne dieses Werkzeug auskommen sollte. Für den ernsthaften Geos-Anwender ist vor allem die RAM-Erweiterung 1750 eine Anschaffung, die sich vom ersten Augenblick an bezahlt macht. Da bei maximal zwei verwalteten Laufwerken nur eine RAM-Disk ermöglicht wird, aber die 1750-Erweiterung damit noch lange nicht ausgelastet ist, bietet sie noch weitere Vorteile von unschätzbarem Wert.

So wird ein anderes Laufwerk wahlweise »schattiert«; dies bedeutet, daß so viele Lese-Operationen wie möglich über die RAM-Disk ablaufen und oft benötigte Teile einer Diskette somit schneller verfügbar sind. Dennoch bleiben alle Vorteile einer »echten« Diskettenstation erhalten; Sie brauchen sich also um nichts weiter zu kümmern, um die Vorteile der Schattierung zu genießen.

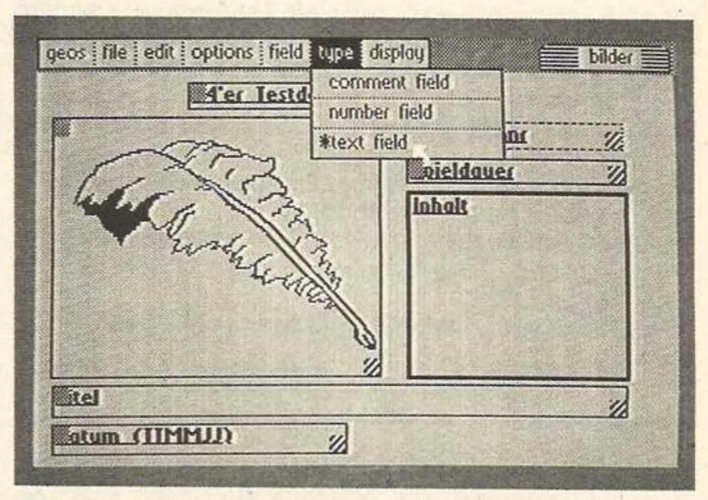


Bild 9. GeoFile, die Datenbank mit Grafikfähigkeit

Zusätzlich beschleunigt die 1750 alle Speicherverschiebungen in großem Stil; das Bewegen in einem GeoPaint-Dokument wird damit beispielsweise um ein Vielfaches schneller.

Nicht zu verachten ist auch die Option, einen Neustart von Geos (nach Reset oder softwaremäßigem Verlassen von Geos) mit der RAM-Disk-Geschwindigkeit ausführen zu lassen; man gibt einfach LOAD"RBOOT",8,1 ein, und maximal 17 Sekunden später ist Geos wieder verfügbar (das normale Booten über LOAD ":*",8,1 dauert mehr als doppelt so lang). Der Inhalt einer RAM-Disk bleibt in diesem Ausnahmefall erhalten.

Und so funktioniert's

Abschließend soll den Floppy-Freaks unter Ihnen noch ein Überblick darüber vermittelt werden, wie Geos seine Disketten organisiert. Zum Verständnis dieser Erläuterungen sollten Sie bereits den Aufbau einer herkömmlichen C 64/128-Diskette kennen; dieser wird im Kurs »In die Geheimnisse der Floppy eingetaucht« an anderer Stelle in diesem Sonderheft erläutert.

Zunächst fällt bei Betrachtung von Geos-Disketten über LOAD"$",8 und LIST auf, daß teilweise nur Grafikzeichen in den Dateinamen stehen; dies liegt daran, daß Geos sich einer anderen Zeichencodierung als der C 64/128 bedient. Im Kleinschriftmodus des C 64/128 erkennt man jedoch zumindest die Dateinamen, wobei dann die Groß- und Kleinbuchstaben vertauscht sind.

Im wesentlichen liegen bei Geos die Großbuchstaben auf den Codes $41 bis $5a und die Minuskeln (Kleinbuchstaben) auf $61 bis $7a; im Kleinschriftmodus des C 64/128 liegen die Minuskeln auf $41 bis $5a, die Majuskeln hingegen auf $c1 bis $da oder $61 bis $7a.

Zahlen und Satzzeichen verändern ihre Codes nicht.

Somit sind Geos-Disketten mit gewöhnlichen C 64-Diskettenmonitoren nicht zu bearbeiten. Mittlerweile gibt es jedoch zwei Standardprogramme zu diesem Zweck: Der »Geos-Disk-Monitor«, kurz »GDM« genannt, welcher im 64'er-Geos-Kurs (Ausgabe 2/87) veröffentlicht wurde, und

den GeosMON, eine umgeschriebene Variante des SMON, welche sich auf der Diskette befindet, die den beiden am Ende des Artikels angegebenen Büchern beiliegt.

Diese Programme konvertieren selbständig die Geos-Zeichen ins C 64-Format und lösen somit alle Codierungsprobleme.

Die BAM unter Geos

Der Block 18/0 beinhaltet den Blockbelegungsplan (BAM) über eine Diskette sowie Disketten-Namen und -ID. Geos nutzt den noch ungenutzten Raum teilweise für eigene Informationen:

- In den Bytes 174-189 steht der Text »Geos format V1.0« im Geos-Code. Diese Kennung wird bei Öffnen einer Diskette abgefragt.
- Die Bytes 172/173 geben Spur und Sektor des Blockes an, welcher die acht Dateien des Disketten-Randes (Border) enthält. Legt der Anwender eine Datei auf den Rand, so wird sie im eigentlichen Directory gelöscht und in den Border-Block eingetragen, welcher vom Format her nur ein weiterer Directory-Block (außerhalb von Spur 18) ist.
- Der Diskettentyp ($42 = Systemdiskette, $50 = Hauptdiskette, andere Werte = Arbeitsdisketten) wird in Byte 190 (von 1 an gezählt) aufbewahrt.
- Die Bytes 190-256 bleiben weiterhin ungenutzt.

Directory-Einträge

Jede Datei wird im Directory auf Spur 18 vermerkt. Verfolgt man dessen sequentielle Verkettung, fällt einem vielleicht auf, daß auf 18/1 nicht 18/4, sondern 18/9 folgt. Dies liegt daran, daß der sogenannte »Interleave« (der physikalische Abstand der logischen Sektoren) bei Geos nicht 3 (4-1), sondern 8 (9-1) beträgt. Der Geos-Floppy-Speeder kommt nämlich mit diesem anderen Blockabstand besser - sprich: schneller - zurecht.

Nun zu den Einträgen der einzelnen Dateien. Auch hier bleibt festzustellen, daß Geos neben den bereits vom DOS getätigten Vermerken auch eigene Informationen ablegt:

- Die Bytes 22/23 (wieder von 1 an gezählt) in einem Eintrag (jeweils 32 Byte Länge) geben Spur und Sektor des sogenannten Info-Blockes an; dieser Block enthält weitere Informationen und gehört ebenso zur gesamten Datei wie die Datenblöcke selbst. Näheres dazu folgt.
- Die Filestruktur steht in Byte 24 (0 = sequentiell, 1 = VLIR).
- Der Geos-Filetyp hat eine Kennziffer von 0 bis 14 und steht in Byte 25 (0 = C»C= PRG« usw.).
- Das Datum ist in den Bytes 26 bis 30 enthalten; diese beziehen sich auf den Zeitpunkt der letzten Änderung: 26 = Jahr (0-99, also ohne Jahrhundertangabe), 27 = Monat (1-12), 28 = Tag (1-31), 29 = Stunde (0-24), 30 = Minute (0-59).

Der Infosektor

Der angesprochene Infoblock (siehe Byte 22/23) enthält 256 weitere Byte voller Informationen über Geos. Diese Daten gehören nicht zum eigentlichen Datei-Inhalt; sie stehen als zusätzlicher, in der BAM belegter Block auf Diskette und werden der Einfachheit halber bei der Dateilänge mitgerechnet.



Bild 10. Die RAM-Erweiterungen 1750 und 1700

Folgende Daten enthält ein Infoblock:
- Sprite-ähnliches Format des Datei-Piktogramms (Icon, Kleingrafik)
- noch einmal zur Kontrolle: Filetyp und Struktur
- Anfangs- und Endadresse im Speicher sowie die absolute Startadresse
- »Klasse« und »Autor«











- bei Datenfiles: der Name des Anwendungsprogramms, welches bei Doppelklicken der Datei einzulesen ist
- Informationstext

Bislang haben wir »von außen« betrachtet, wie Geos mit der Floppy umgeht. Abschließend sei nur kurz berichtet, wie man in eigenen Geos-Programmen die Floppy ansteuert.

Die in Tabelle 1 aufgelisteten Routinen stehen, ähnlich den Kernel-Routinen des C64, ausschließlich für Floppy-Operationen zur Verfügung (in Klammern die Adressen der Sprungtabelle, welche sowohl für alle C64-Versionen von Geos als auch Geos 128 zutreffen).

Aus dieser Unmenge an Routinen können Sie ersehen, daß deren Beschreibung aus Platzgründen hier nicht möglich ist. Deshalb haben Sie sicher Verständnis, wenn ich Sie im Literaturverzeichnis auf zwei Bücher zu Geos hinweise, welche alle Geos-Routinen und das Floppy-Format ausführlich beschreiben. Auf der dem Buch beiliegenden Diskette befinden sich Beispieldateien, -programme und viele Utilities zur Floppy-Behandlung (Geosmon, Geos Icon-Editor, Geos File-Analyzer, Geos File-Freak, Geos VLIR-Analyzer und viele mehr).

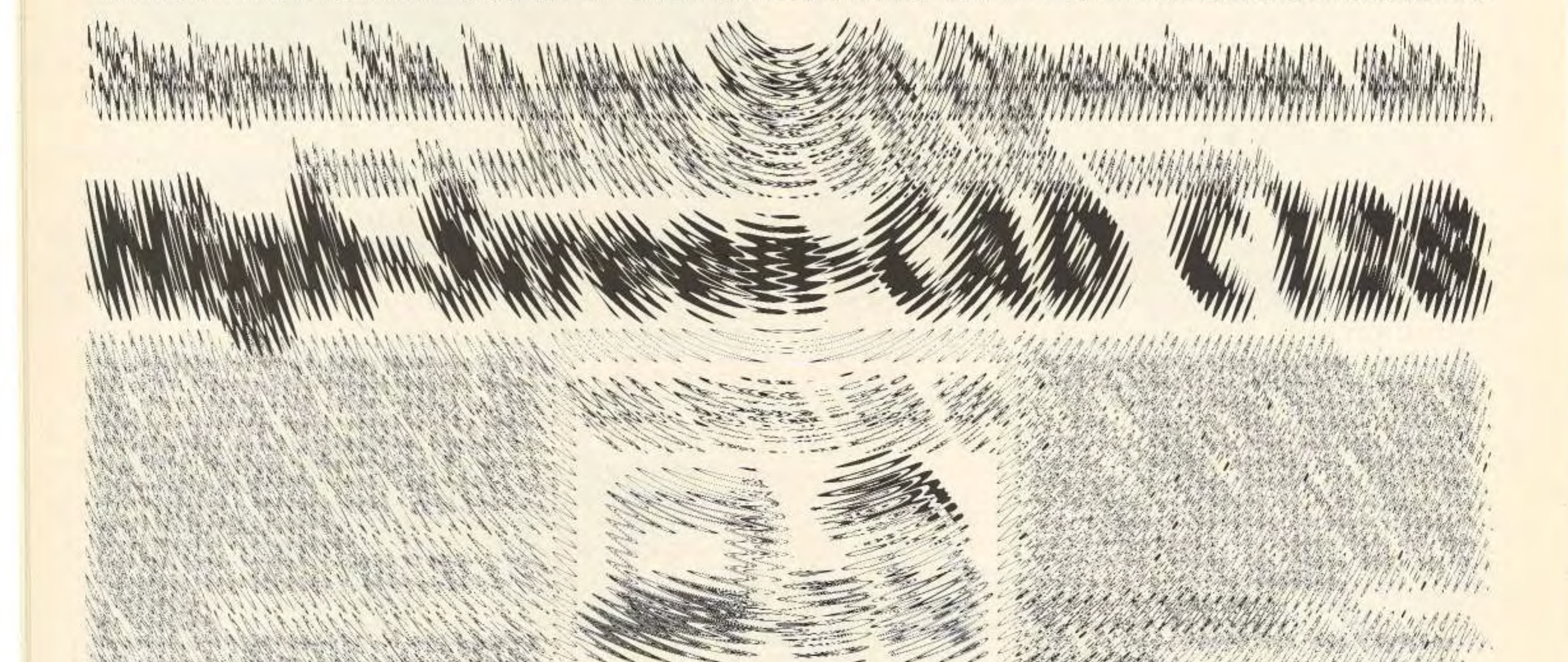
Ebenfalls interessant ist der Geos-Programmierkurs im 64'er-Stammheft (Ausgaben 2/87-9/87), welcher jedoch die Thematik »Floppy und Floppy-Programmierung« nur teilweise anspricht. (Florian Müller/sk)

Floppy-Routinen in Geos:		
LOADSET ($c1cc)	BLKFREE ($c1db)	CHKDISK ($c1de)
NEWDISK ($c1e1)	GETBLK ($c1e4)	PUTBLK ($c1e7)
SETDISK ($c1ea)	SAVE ($c1ed)	SETDIR ($c1f0)
BLDDIR ($c1f3)	GETFREE ($c1f6)	WRITE ($c1f9)
ALLOBLK ($c1fc)	READ ($c1ff)	CHAIN ($c205)
GETFILE ($c208)	FINDFILE ($c20b)	CRC ($c20e)
LDFILE ($c211)	TURBON ($c214)	LDDESK ($c217)
READBLK ($c21a)	LDAPPL ($c21d)	WRITEBLK ($c220)
VERWRITE ($c223)	FREEFILE ($c226)	GETHDR ($c229)
DESKTOP ($c22c)	TURBOFF ($c232)	TURBOPUR ($c235)
DELETE ($c238)	FINDTYPE ($c23b)	RESTART ($c23e)
FASTDEL ($c244)	GETDIR ($c247)	PUTDIR ($c24a)
ALLONXT ($c24d)	INITIO ($c25c)	DONEIO ($c25f)
NEXTFREE ($c292)	UPDATE ($c295)	DRVNAM ($c298)
OPENDISK ($c2a1)	READBYTE ($c2b6)	CHANGE ($c2bc).

Speziell für die Arbeit mit VLIR-Dateien sind folgende Routinen vorgesehen:	
OPENREC ($c274)	CLOSEREC ($c277)
NEXTREC ($c27a)	PREVREC ($c27d)
POINTREC ($c280)	DELREC ($c283)
INSREC ($c286)	APPREC ($c289)
READREC ($c28c),	WRITEREC ($c28f).

Tabelle 1. Die Floppy-Routinen, die Geos bereitstellt, mit ihren Kürzeln und Einsprungadressen in Klammern

Geos-Titel der Commodore Sachbuchreihe:
Florian Müller/Thorsten Petrowski, C64 — Alles über Geos V1.2 (Anwendungs- und Programmierhandbuch), über 530 Seiten, rund 200 Abbildungen, inkl. doppelseitiger Diskette mit Beispielprogrammen und Geos-Utilities, Markt & Technik Verlag, Bestellnummer MT 90461, 49 Mark
In derselben Aufmachung mit einem zusätzlichen Farbteil, aber mit völlig neuem und überarbeiteten Inhalt, ist auch seit Dezember 1987 das Buch »C64 — Alles über Geos V1.3« erhältlich, welches sich speziell mit der deutschen Programmversion Geos 1.3 befaßt.
Geos im 64'er-Magazin:
Kurs: Der Schlüssel zu Geos, Ausgaben 2/87-9/87. Behandelt wird die Programmierung von Geos in Assembler.
Neuer Kurs ab Ausgabe 12/87. Dieser Kurs gliedert sich in einen reinen Einsteigerteil sowie eine parallel dazu erscheinende Tips-und-Tricks-Ecke.
Geos V1.3 und Geos 128 sowie die Geos-Applikationen erhalten Sie im Fachhandel sowie beim Markt & Technik Verlag, Hans-Pinsel-Str. 2, 8013 Haar bei München, Tel. 089/4613-0



Informationen im Quartett

Floppystationen sind als Massenspeicher für den C64 oder C128 kaum wegzudenken. Entsprechend groß ist der Informationsbedarf rund um diese Geräte. Um diesem nachzukommen, veröffentlicht das 64'er-Magazin nun bereits das vierte Sonderheft zum Thema »Floppy-Laufwerk«.

Nahezu jeder Besitzer einer Datasette hegt früher oder später einmal den Wunsch, eine Floppystation zu besitzen. Denn die Datenaufzeichnung auf Kassetten ist langsam und unkomfortabel. Oftmals verstreichen viele Minuten, bis sich ein Programm im Speicher des C64 oder C128 befindet. Um ein bestimmtes Programm auf einer Kassette zu finden, muß meist die gesamte Kassette durch langwieriges Vor- und Zurückspulen abgesucht werden, insbesondere wenn man sich den Zählwerkstand der einzelnen Programme nicht notiert hat. Zudem ist der Gebrauch der Datasette unsicher. Schadhaftes Bandmaterial der Kassetten führt nicht selten zu Aufzeichnungs- und Lesefehlern.

Angesichts dieser Nachteile machten viele Besitzer eines C64 oder C128 den Traum von einer Floppy 1541 oder 1571 wahr, um die bequeme Datenaufzeichnung auf Disketten genießen zu können. Heute ist das Floppylaufwerk aus dem Leben eines C64-Anwenders nicht mehr wegzudenken. Eine Diskettenstation ist jedoch, wie der Computer selbst, ein sehr komplexes Gerät, das viele Geheimnisse in sich birgt.

Seit Beginn der C64-Ära hilft deshalb das 64'er-Magazin, das »Mysterium« um die Floppystation zu beseitigen. Bald schon konnten komplette Sonderhefte mit Tips und Programmen für die Floppy 1541 und 1571 gefüllt werden. Nun halten Sie bereits die vierte Sonderausgabe in den Händen, die sowohl dem Einsteiger als auch dem fortgeschrittenen Computer-Anwender interessante und unerläßliche Informationen über das Speichermedium »Floppystation« gibt.

Obgleich es sich auch bei dieser Ausgabe um ein in sich abgeschlossenes Sonderheft handelt, kann es doch als Fortsetzung der anderen drei Sonderhefte verstanden werden. Denn diese haben kaum an Aktualität eingebüßt. Dies wollen wir Ihnen nun zeigen, wenn wir eine Rückblende auf die vorangegangenen Sonderhefte 5/85, 9/86, und 5/87 unternehmen.

Aller Anfang ist schwer

Mitte 1985, als der C64 gerade begann, Fuß zu fassen, war die Datasette nicht zuletzt wegen ihres günstigen Preises ein noch weit verbreitetes Speichermedium. Nur wenige Freaks konnten sich an einer Floppy 1541 erfreuen. So widmete sich das erste Sonderheft, das sich mit der Floppy 1541 beschäftigte, nicht nur ausschließlich diesem Thema. Das Sonderheft 5/85 enthielt somit auch Basic-Erweiterungen, Tips und Tricks zur Datasette und anderen Bereichen. Doch bereits hier zeichnete sich bei der Auswahl der Programme für die Diskettenstation der Trend nach Professionalität ab.

So findet man in Sonderheft 5/85 unter den Basic-Erweiterungen gleich zwei Programme mit den Namen »Disk-Basic« und »Disk-Basic 64«, die die Arbeit mit der Floppystation erleichtern sollen. Spezielle Anweisungen ähnlich denen des C128, dienen der vereinfachten Handhabung der Floppy 1541. Beispiele sind hier FORMAT, CATALOG, COLLECT oder SCRATCH.

Eine weitere Gruppe von Anweisungen, wie DOPEN, FETCH oder RECORD, erlaubt es dem Anwender, sequentielle und relative Dateien auf der Diskette schnell und problemlos zu verarbeiten, ohne direkt auf den Kommandokanal zugreifen zu müssen. Zudem lassen sich Dateien auf einfache Weise miteinander verketten, auf andere Peripheriegeräte transferieren oder vor unbeabsichtigtem Löschen schützen. Disk-Basic 64 besitzt gar einen Befehl (RESCUE), der gelöschte Dateien wieder rekonstruiert.

```
           **** f-mon 1541 ****
      (c) 1985 by m.koehler p.baumann

.m
.:0300 12 07 82 1b 01 52 45 43 .....rec
.:0308 4f 52 44 2d 42 45 46 2e ord-bef.
.:0310 a0 a0 a0 a0 a0 00 00 00      ...
.:0318 00 00 00 00 00 00 04 00 ........
.:0320 00 00 80 12 01 2d 2d 2d .....---
.:0328 2d 2d 2d 2d 2d 2d 2d 2d --------
.:0330 2d 2d 2d 2d 2d 00 00 00 -----...
.:0338 00 00 00 00 00 00 00 00 ........
.:0340 00 00 82 09 05 43 4f 44 .....cod
.:0348 45 20 46 52 45 45 a0 a0 e free
.:0350 a0 a0 a0 a0 a0 00 00 00      ...
.:0358 00 00 00 00 00 00 1a 00 ........
.:0360 00 00 80 12 01 2d 2d 2d .....---
.:0368 2d 2d 2d 2d 2d 2d 2d 2d --------
.:0370 2d 2d 2d 2d 2d 00 00 00 -----...
.:0378 00 00 00 00 00 00 00 00 ........
.:0380 00 00 82 0f 01 50 52 49 .....pri
.:0388 4e 54 20 55 53 49 4e 47 nt using
.:0390 a0 a0 a0 a0 a0 00 00 00      ...
.
```

Bild 1. FMON, der ungewöhnliche Floppy-Monitor ist gerade beim Disassemblieren eines Maschinenprogramms

Weniger für Basic-Kenner als für die eingefleischten Floppy-Freaks ist ein weiteres Hilfsprogramm mit dem Namen »FMON 1541« gedacht. Ähnlich dem Programm »SMON« für den C64 handelt es sich bei FMON um einen Maschinensprache-Monitor. Er ist jedoch kein Monitor im herkömmlichen Sinne, da er sich nicht mit der Maschinensprache des Computers befaßt. Er dient vielmehr der professionellen Programmierung der Floppy 1541. Denn FMON erlaubt den komfortablen Zugriff auf das RAM und ROM der Diskettenstation, das der Anwender bisher nur unter Mühen erreichen konnte.

Mit einem Mini-Disassembler erhält man die Gelegenheit, den gesamten Speicherbereich sowie das DOS, das heißt das Betriebssystem der Floppy 1541, als Assemblerprogramm zu betrachten. Analog dazu erlaubt ein Direkt-Assembler, wie man ihn in Bild 1 in Aktion sehen kann, die bequeme Eingabe von Maschinenprogrammen in den RAM-Bereich der Floppystation, um sie dort später ablaufen zu lassen. Eine Besonderheit dieses Assemblers ist es, daß er anders als die üblichen Direkt-Assembler auch »nichtdefinierte Befehle« (illegale Opcodes) der Maschinensprache verarbeitet.

Sicher ist sicher

Ob Sie nun ein Programm mit FMON, SMON oder nur einfach in Basic geschrieben haben, ist es in jedem Fall wünschenswert, daß keine unbefugten Augen Ihr so mühsam entwickeltes Werk einsehen können. Ein Programmschutz ist hier sehr nützlich. Haben Sie bisher noch keine wirksame Lösung gefunden, bietet das Sonderheft 5/85 eine raffinierte Variante, Programme vor unberechtigter Nutzung zu

schützen. Das Programm mit dem Namen »Autostart C 64« versieht zunächst jedes beliebige Basic- oder Maschinenprogramm mit einem Autostart. Das bedeutet, daß die Programme sofort nach dem Laden von Diskette starten, ohne

```
 *  *  *  Autostart C64 Plus  *  *  *
Enter Sourcefilename? autostart
Enter Targetfilename? autostart 2

Basicprogramm  (j/n)? j


Restoreprotect (j/n)? j
Stopprotect    (j/n)? j
Codeword            ? 1987

Entery correct (j/n)?
```

Bild 2. »Bitte Paßwort eingeben«. Autostart 64 läßt keinen Unbefugten an Ihre Programme heran.

dem Anwender die Möglichkeit zu geben, die Programmzeilen mit LIST oder einem Monitor einzusehen. Diese sehr effektive Methode ist nur schwer rückgängig zu machen. Sie alleine wäre jedoch kein ausreichender Schutz, denn die so modifizierten Programme sind nach wie vor einfach kopierbar.

Aus diesem Grund hält »Autostart C 64« ein weiteres Hindernis parat, das »Software-Klauern« das Handwerk legen soll. Eine interessante Verschlüsselungsmethode verwandelt jeden Befehl des zu schützenden Programms in ein unleserliches Wirrwarr von Zeichen und Zahlen. Nur ein Paßwort kann das funktionsunfähige Durcheinander wieder zu einem ablauffähigen Programmcode formen. Dieses ist vom Anwender bei der Verschlüsselung frei wählbar, und wird später vor dem Start des geschützten Programms abgefragt (Bild 2). Wurde das Paßwort richtig eingetippt, entwirrt sich das codierte Programm in seinen ursprünglichen Zustand und startet schließlich automatisch.

Auf diese Weise haben nur Personen, die das entsprechende Geheimwort kennen, Zugang zu Ihrem Werk. Die Schutzmaßnahmen von »Autostart C 64« sind so einfach wie wirksam, haben jedoch einen entscheidenden Nachteil. Man muß sich jedes der verwendeten Paßwörter fest einprägen. Denn sollten Sie ein Paßwort vergessen, besteht auch für Sie selbst keine Möglichkeit mehr, auf Ihr geschütztes Programm zuzugreifen. Wer also von chronischer Vergeßlichkeit geplagt ist, muß sich mit »Eselsbrücken« oder Paßwort-Schemata helfen.

Die elektronische Schreibschutzplakette

So einfallsreich man ein Programm vor unbefugter Benutzung oder unerlaubtem Kopieren auch gesichert haben mag, so droht den Daten dennoch eine große Gefahr. Durch fahrlässige Verwendung des SCRATCH-Befehls kann eine Datei unbeabsichtigt oder mutwillig von einer Diskette gelöscht werden. Um dies zu vermeiden, genügt in der Regel das Überkleben der Schreibschutzkerbe, die sich im Mantel der Diskette befindet. Oftmals lösen sich die Klebestreifen jedoch wieder, oder sie werden bewußt entfernt. Die Daten stehen dem Löschen wieder schutzlos gegenüber.

Wesentlich eleganter und sicherer ist es, einen softwaremäßigen Schreibschutz auf die Disketten aufzubringen. Ein kleines Programm namens »Schreibschutz« in Sonderheft 5/85 ist Ihnen dabei behilflich. Auf einfache Weise wird jede beliebige Diskette mit einem veränderten Formatkennzeichen versehen. Solch präparierte Disketten haben die Eigenschaft, daß die Floppy 1541 oder 1571 jeden Schreibzugriff darauf verweigert. Jegliche Lösch- und Speicherversuche mit SCRATCH oder SAVE sind vergebens.

Selbstverständlich kann der Schreibschutz bei Bedarf jederzeit wieder aufgehoben werden, so daß der Zugriff auf die Daten wieder uneingeschränkt möglich ist.

Leider schützt die »Software-Schreibschutzplakette« nicht vor einer Neu-Formatierung einer Diskette. Hier hilft nur mehr das herkömmliche Klebeband.

Floppystation exklusive...

Das Sonderheft 9 der 64'er-Zeitschrift ist die zweite Sonderausgabe, die sich intensiv mit der Floppy 1541 und 1571 beschäftigt. Im Gegensatz zu Sonderheft 5/85 ist es jedoch ausschließlich diesem Thema gewidmet.

Ein umfassender Hardware-Test gibt zunächst Auskunft über die gängigen Diskettenstationen von Commodore. In einer Gegenüberstellung werden die Floppies 1541, 1570/71 und 1551, die man in Bild 3 sehen kann, miteinander verglichen. Vor allem Einsteiger, die ein Floppylaufwerk erstehen wollen, erfahren hier, welches der verschiedenen Modelle für ihren Computer am besten geeignet ist. Wer bereits im Besitz einer Floppystation ist, wird diesen Bericht ebenfalls mit großem Interesse verfolgen, da er wesentliche Unterschiede der Floppy-Modelle aufdeckt. Die gewonnenen Informationen können beispielsweise dazu verwendet werden, Programme an andere Floppystationen anzupassen oder die Eigenheiten des Gerätes tiefer zu ergründen.

So unterschiedlich die verschiedenen Diskettenlaufwerke von Commodore auch sein mögen, eine grundlegende Eigenschaft ist ihnen jedoch gemein. Obgleich sie wesentlich schneller arbeiten als die Datasette, ist ihre Lese- und Schreibgeschwindigkeit im Vergleich zu den Diskettenstationen anderer Computersysteme sehr langsam. Dies ist jedoch nicht, wie man vermuten könnte, der unzureichenden Leistung der Floppystationen zuzuschreiben. Jedes Floppylaufwerk ist normalerweise imstande, Daten in enormen Geschwindigkeiten zu lesen oder zu schreiben. Der Engpaß liegt vielmehr an der Schnittstelle zum Computer. Die Daten werden nämlich über ein langsames serielles Kabel gesendet. Lediglich die Floppy 1551 für die Computer C 16 und Plus/4 besitzt eine parallele Schnittstelle, was zu einer schnelleren Datenübertragung führt.

Der ermüdenden Geschwindigkeit der Floppy 1541 überdrüssig, fanden findige Leser des 64'er-Magazins zunächst Möglichkeiten, den Datentransfer über das serielle



Bild 3. Ein Turm von verschiedenen Floppymodellen für Commodore-Computer baut sich vor dem Einsteiger auf. Welche ist die richtige für Ihren Computer?

Kabel durch geeignete Programme zu optimieren. Den Anfang machten legendäre Programme wie »Hypra-Load« (Ausgabe 10/84, 64'er) und »Hypra-Save« (Ausgabe 5/86, Sonderheft 64'er), die das Laden und Speichern von Programmen um das Fünf- bis Sechsfache beschleunigen. Den bisherigen Rekord brach das Betriebssystem »Exos V3«, das in Ausgabe 12/86 des 64'er-Magazins vorgestellt wurde. Lediglich über die serielle Schnittstelle gelingt es Exos, Daten in einer Geschwindigkeit zu laden, die um das 14fache der ursprünglichen Geschwindigkeit liegt. Die »Schallgrenze« der seriellen Übertragung scheint damit endgültig erreicht.

Sollen Daten noch schneller gelesen werden, ist bereits zusätzliche Hardware nötig. Mit geeigneten Erweiterungsplatinen für Computer und Floppystation und der wesentlich schnelleren Datenübertragung per Parallelkabel werden neue Dimensionen der Datenspeicherung erreicht. Leistungsfähige »Floppy-Speeder«, wie jene Hardware-Erweiterungen auch genannt werden, erhöhen die Arbeitsgeschwindigkeit der Floppystation nicht selten um das 30- bis 65fache (siehe auch Speeder-Test in dieser Ausgabe).

Daten verwalten wie ein Profi

Haben Sie unter Zuhilfenahme der beiden Testberichte die richtige Floppystation und den besten Hardware-Speeder gefunden, können Sie sogleich beginnen, die Vorteile eines Diskettenlaufwerks zu nutzen. Sie könnten zum Beispiel ein Verwaltungsprogramm für die Adressen Ihrer Freunde programmieren oder gar mit einer Schallplattendatei endlich Ordnung in Ihre Musiksammlung bringen. Wenn Sie mit der Speicherung von Daten auf Diskette noch nicht ganz vertraut sind, um sich an solche Projekte zu wagen, schlagen Sie doch einfach Sonderheft 9 auf. Denn dort kann man in einem ausführlichen Grundlagenartikel alle nötigen Informationen zur Dateiverwaltung erfahren.

Der Bericht beschäftigt sich zunächst mit der einfachsten Art von Dateien, den sequentiellen Files. Komplizierter, aber wesentlich effektiver sind relative Dateien, die den direkten Zugriff auf bestimmte Datenelemente erlauben. Die Programmierung ist hier bereits etwas umständlich. Ein eigenes Kapitel beschäftigt sich daher nur mit der relativen Datenspeicherung.

Besonders interessant ist die geschickte Verquickung von sequentiellen und relativen Dateien zu »indexsequentiellen Dateien«. Durch sogenannte Indexfelder lassen sich solche Dateien nach bestimmten Kriterien absuchen, um so ein bestimmtes Datenelement zu finden.

Mit den »Random-Access«-Dateien stößt man schließlich in die Welt der professionellen Datei-Programmierung vor. Losgelöst von den festgelegten Formaten der sequentiellen und relativen Files werden Random-Access-Dateien vom Programmierer direkt auf der Diskette verwaltet. Hier ist es dem Anwender überlassen, wie er die einzelnen Blöcke einer Diskette organisiert, um möglichst schnell und vorteilhaft an gewünschte Daten zu gelangen.

Nach Studium dieses Artikels besitzen Sie nun alle Voraussetzungen, Ihre eigene Dateiverwaltung auf dem C 64 oder C 128 zu entwerfen. Haben Sie jedoch nicht die Muße, ein solches Projekt in Angriff zu nehmen, bietet Sonderheft 9 quasi als praktisches Beispiel ein interessantes und flexibles Programm zur Bearbeitung von umfangreichen Daten.

Ob Sie nun wichtige Adressen oder die Filmtitel Ihrer Video-Sammlung zu verwalten haben, überall wo Ordnung von großer Wichtigkeit ist, kann »Datec«, ein professionelles Datenverarbeitungs-Programm, wertvolle Dienste leisten. Bis zu 1024 beliebige Datensätze mit einer maximalen Länge von 256 Zeichen können mit Datec problemlos und sicher verwaltet werden.

Die Eingabe der Daten erfolgt bequem über eine Bildschirm-Maske, die man nach eigenen Vorstellungen gestalten kann. Bild 4 zeigt zum Beispiel die mögliche Eingabe-Maske für eine Adreßverwaltung. Einfache Kommandos gestatten es, die eingegebenen Datensätze auf Tastendruck vor- und zurückzublättern, zu ändern, zu löschen oder gar auf einem angeschlossenen Drucker auszugeben.



Bild 4. Eine Eingabemaske von Datec kann nach Belieben gestaltet werden. Hier zum Beispiel für eine Adreßdatei.

Werden Programme oder andere Dateien auf einer Diskette nicht mehr gebraucht, kann man bekanntlich die gesamte Diskette durch einen Formatiervorgang vollständig löschen. Alle darauf befindlichen Bytes werden dadurch überschrieben.

Sollen hingegen nur einzelne Dateien von der Diskette entfernt werden, bedient man sich in der Regel des Floppy-Befehls »S« oder »SCRATCH«, der über den Kommando-Kanal der Floppystation gesendet werden muß. Nach dem Löschvorgang scheint es, als wäre das betreffende Programm tatsächlich verschwunden, da es im Inhaltsverzeichnis der Diskette nicht mehr aufgeführt wird. Nur wenige wissen aber, daß mit dem »SCRATCH«-Befehl lediglich der Eintrag im Directory verändert wird, das Programm hingegen unbeschädigt auf der Diskette verbleibt, bis es von anderen Daten überschrieben wird.

Löschen ohne Wenn und Aber

Dieser Tatsache ist es zu verdanken, daß man mit etwas Geschick und einem Diskettenmonitor eine durch Scratch verloren geglaubte Datei wieder rekonstruieren kann. Wer einmal ein Programm versehentlich gelöscht hat, weiß dies zu schätzen.

Sind jedoch wichtige Daten zu löschen, wie zum Beispiel eine persönliche Datec-Adreßkartei, besteht die Gefahr, daß unbefugte Hände die angeblich eliminierte Datei wiederherstellen. Im Sinne des Datenschutzes ist es deshalb nötig, Programme und Dateien mitunter vollständig zu löschen. Mit »Physikal Scratcher«, einem kleinen Maschinenprogramm aus Sonderheft 9, ist dies problemlos zu bewerkstelligen, indem alle Blöcke, die von der zu löschenden Datei belegt werden, mit Null-Bytes überschrieben werden. Die Daten sind damit endgültig gelöscht. Bei der Anwen-

dung dieses Löschprogramms ist jedoch große Vorsicht geboten, da die behandelten Dateien nun tatsächlich nicht mehr rekonstruierbar sind.

Luxus-Betriebssystem für den C 64

Als eifriger Leser des 64'er-Magazins ist Ihnen sicherlich das Betriebssystem »64'er-DOS« für den C 64 ein Begriff. In der 64'er-Ausgabe 3/86 erstmals vorgestellt, erlangte es unter den C 64-Besitzern bald große Bekanntheit, da es wesentliche Erleichterungen im Umgang mit dem Computer und der Floppystation bot. Zudem ist das »64'er-DOS«, wenn es auf EPROMs gebrannt und anstelle des Original-Kernels in den C 64 eingesetzt wird, sofort nach dem Einschalten des Computers aktiv, so daß man gleich die angenehmen Eigenschaften dieses hervorragenden Betriebssystems genießen kann. Das hauptsächliche Ziel von 64'er-DOS ist es, die Arbeit mit Computer und Diskettenstation so bequem wie möglich zu gestalten. Verschiedenste Hilfsfunktionen, die der Anwender per Tastendruck aufrufen kann, gestatten beispielsweise das Anzeigen des Directories ohne Programmverlust oder die Abfrage des Fehlerkanals der Floppy 1541, während neue Floppy-Routinen ein schnelleres Laden und Speichern von Daten garantieren.

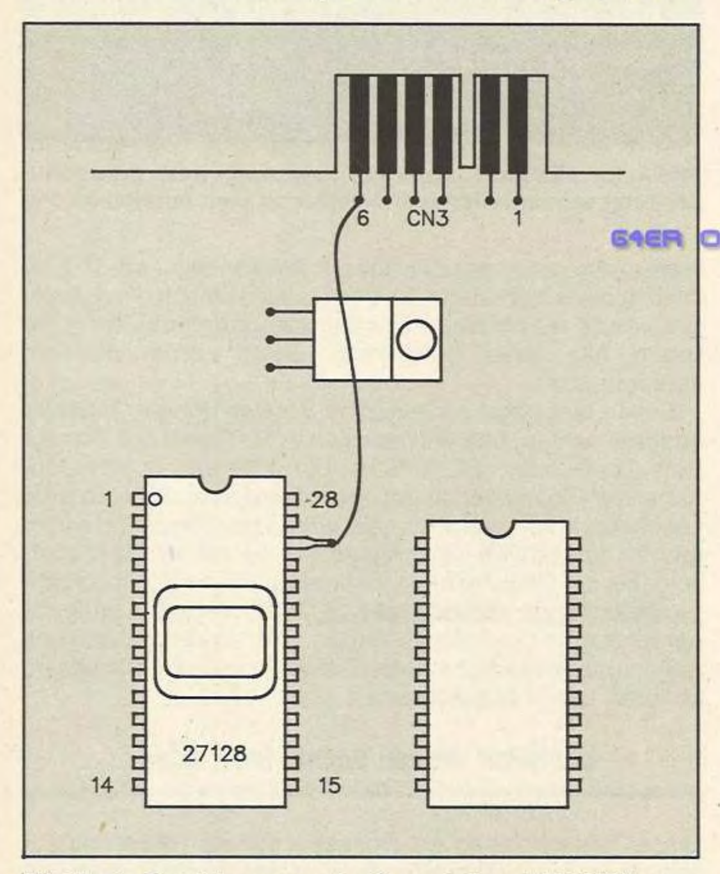


Bild 5. Ein Betriebssystem der Superlative: »EX-SMON-DOS« mit einem eingebauten Maschinensprache-Monitor

In Sonderheft 9 findet man nun eine wesentlich erweiterte Version von 64'er-DOS, die mit einer Fülle von zusätzlichen Programmierhilfen ausgerüstet ist. Hat man »EX-SMON-DOS«, wie das neue 64'er-DOS getauft wurde, auf EPROMs gebrannt und in die Floppy 1541 und den C 64 eingesteckt, stehen ab sofort neben den Annehmlichkeiten des 64'er-DOS zusätzlich ein umfassender Monitor, viele Basic-Hilfsfunktionen und sogar eine Hardcopy-Routine zur Verfügung.

Die Bedienung der Erweiterungen ist denkbar einfach. Durch Voranstellen des Zeichens ».« gelangt man zum Beispiel direkt vom Basic-Editor in den komfortablen Maschinensprache-Monitor, der ähnlich leistungsfähig wie der bekannte Monitor SMON ist (Bild 5).

Über weitere Tastenkombinationen erreicht man die Basic-Tools, die dem Basic-Programmierer das Leben erleichtern sollen. Dazu gehört etwa die automatische Zeilennumerierung, das komfortable Löschen von Programmzeilen oder das Anzeigen eines Variablen-Dumps. Mit dem Kommando MOVE können sogar Gruppen von Zeilen innerhalb eines Programms verschoben werden.

EX-SMON-DOS zeigt, wie komfortabel und anwenderfreundlich ein Betriebssystem aussehen kann. Wer EX-SMON-DOS einmal kennengelernt hat, wird es bald nicht mehr missen wollen.

Selbstverständlich ist mit EX-SMON-DOS die Programmfülle von Sonderheft 9 noch lange nicht erschöpft. Weiterhin finden sich ein schneller Algorithmus zum Sortieren von Daten, ein leistungsfähiges Kopierprogramm für zwei Floppies 1541, ein Programm zum Herstellen von Diskettenhüllen mit Inhaltsverzeichnis und vieles mehr.

Viele Informationen

Betrachten wir nun Sonderheft 15, das sich thematisch nahtlos an Sonderheft 9 anschließt, fällt uns sogleich der große Anteil an umfassenden Grundlagenartikeln auf, die uns die Geheimnisse der Floppy 1541 und 1571 näherbringen sollen.

Gleichgültig, welches Speichermedium Sie verwenden, eine Floppystation und auch die Datasette bedürfen einer gewissen Pflege, um ihre Aufgabe — das Schreiben und Lesen von Daten — stets korrekt verrichten zu können. Ein Bericht in Sonderheft 15 befaßt sich deshalb mit der richtigen Behandlung dieser beiden Geräte. Ausführlich und verständlich erfahren Sie, was bei der Benutzung von Diskettenlaufwerk und Datasette vermieden werden soll, um sie nicht unnötig zu belasten. Dazu gehört beispielsweise der sorgsame Umgang mit den mechanischen Teilen, die gerade bei der Floppystation extrem empfindlich sind, sowie die regelmäßige und vorsichtige Reinigung der Schreib-/Leseköpfe. In diesem Zusammenhang lernen Sie alles über Eigenschaften von Reinigungskassetten und -disketten. Denn diese können bei sorgloser Anwendung enorme Schäden anrichten.

Selbst ist der Computer-Freak

Sollte Ihre Diskettenstation trotz intensiver Pflege einmal nicht richtig funktionieren, dürfen Sie nicht gleich verzagen. Denn oftmals handelt es sich nur um kleinere Mängel, die sich mit etwas Fingerspitzengefühl und einigen Werkzeugen, wie etwa jene in Bild 6, selbst beheben lassen und somit den Gang zum Kundendienst überflüssig machen.

Wer jedoch mit den Interna der Floppy 1541 und 1571 noch nicht vertraut ist, sollte den kleinen Erste-Hilfe-Kurs zur Diskettenstation in Sonderheft 15 studieren. Er enthält nützliche Tips, wie man der Ursache von Fehlfunktionen auf die Spur kommt. Dabei kommt man nicht umhin, das Innenleben der Floppystation zu ergründen. Mit vielen Farbfotos wird auf die Eigenheiten der Laufwerksmechanik und der elektronischen Bauteile aufmerksam gemacht, so daß man nach Absolvierung dieses Floppy-Lehrgangs nicht nur zu einem guten Hardware-»Chirurgen« geworden ist, sondern gleichzeitig auch viel über die interne Arbeitsweise der Diskettenstation gelernt hat.

Mittlerweile gibt es eine Unzahl von Diskettenutilities, die das Arbeiten mit einem Floppylaufwerk mehr oder minder

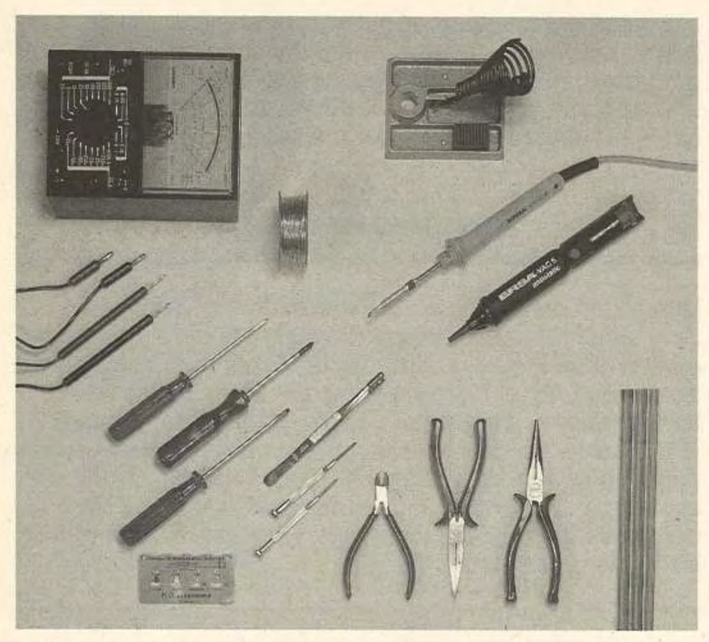


Bild 6. Wer seine Floppystation selbst reparieren will, benötigt verschiedene Werkzeuge und Sonderheft 15

erleichtern. Leider handelt es sich meist um voneinander unabhängige Programme, die stets, wenn dieses oder jenes Utility benötigt wird, einzeln nachgeladen werden müssen. Wie oft hat man sich deshalb schon ein Programm gewünscht, das alle interessanten Diskettenfunktionen in sich vereinigt. Mit dem Programm »Disc-Wizard« aus Sonderheft 15 können wir Ihnen diesen Wunsch mehr als erfüllen. Der Disc-Wizard ist jedoch kein herkömmliches Hilfsprogramm. Viele ungewöhnliche und erstaunliche Funktionen machen ihn zu einem wahren Disketten-Zauberer.

Disketten-Zauber

Gesteuert über ein durchdachtes Menüsystem (Bild 7) lassen sich alle Floppy-Befehle bequem aktivieren, Disketten und einzelne Files mit einem »Software-Schreibschutz« versehen, oder Name und ID einer Diskette problemlos ändern. Daneben können versehentlich gelöschte Dateien oder auch ganze Disketten (kurz formatiert) auf Tastendruck rekonstruiert werden. Die besondere Stärke von Disc-Wizard liegt jedoch in der Manipulation des Inhaltsverzeichnisses einer Diskette. Auf Wunsch können nahezu alle Parameter einer Datei, wie zum Beispiel deren Länge, der Filetyp oder der Namen geändert werden. Daneben dürfen Einträge im Directory nach eigenen Vorstellungen umgestellt und geordnet werden. Bei Bedarf wird das Directory sogar alphabetisch sortiert.

Zudem ist Disc-Wizard auch ein perfekter Disk-Monitor, mit dem die Blockinhalte einer Diskette bequem gelesen und verändert werden können.

Disc-Wizard besitzt damit alles, was man für die angenehme Arbeit mit der Floppystation benötigt, in kompakter Form. Er sollte deshalb in Ihrer Softwaresammlung nicht fehlen.

Der Disketten-Entblätterer

Rein auf die Aufgaben eines Disketten-Monitors hat sich ein weiteres Utility in Sonderheft 9 spezialisiert. »Disk-Mon 64«, so der Name dieses Programms, ist jedoch kein gewöhnlicher Disketten-Monitor, der es nur gestattet, die Inhalte von Blöcken zu manipulieren. Eine Vielzahl von Befehlen ermöglicht es zum Beispiel, Daten einer Diskette direkt als Maschinenprogramm oder Basic-Programm zu interpretieren, Spritedefinitionen zu erkennen oder den Inhalt einer Diskette im ASCII-Format anzuzeigen. Selbstverständlich lassen sich auch direkt Änderungen vornehmen.

Zusätzlich kann eine Diskette nach einer bestimmten Zeichenkette abgesucht, oder beliebige Blöcke mit einem angegebenen Wert gefüllt werden. Keine noch so versteckten Daten sind vor Disk-Mon 64 sicher.

Wer die Ergebnisse seiner Arbeit auf Papier festhalten will, der kann Disk-Mon 64 mit einfachen Kommandos dazu veranlassen, sämtliche Ausgaben, die ursprünglich auf dem Bildschirm erfolgten, auf einen seriell oder parallel angeschlossenen Drucker zu lenken.

Kopieren im Sauseschritt

Schließlich sei noch ein Programm aus Sonderheft 15 erwähnt, das allen C 128-Anwendern, die im Besitz einer Floppy 1570/71 sind, das Herz höher schlagen läßt. Gemeint ist das Kopierprogramm »Thirty Seconds« (Bild 8), das sowohl Disketten mit 1541- und 1571-Format als auch CP/M-Disketten in Windeseile dupliziert. Die Kopierzeit beträgt pro Diskettenseite nur 30 Sekunden. Soll zusätzlich formatiert und die Fehlerfreiheit der geschriebenen Blöcke überprüft werden, erhöht sich die benötigte Zeit lediglich auf 45 Sekunden. Da der C 128 über einen RAM-Speicher von 128 KByte verfügt, sind zudem für eine einseitige Diskette nur mehr drei Diskettenwechsel nötig.

Selbst das Kopieren vieler Disketten wird somit zum Kinderspiel. Nächtelange Kopiersitzungen und der berüchtigte »Diskettenwechsel-Finger« gehören nun endlich der Vergangenheit an.

Neben den eben erwähnten Anwendungs-Programmen enthält das Sonderheft 15 noch eine große Anzahl an Tips und Tricks, die man ebenfalls nicht vergessen sollte. So findet man hier Programme, die nach versteckten Werten auf der Diskette suchen, selbständig einen Diskettenwechsel erkennen oder das Laden von Diskette mittels eines Auswahlmenüs erleichtern. Darüber hinaus erhält man auch ein gutes Kopierprogramm für den C 64, ein Utility zum schnellen Löschen von Dateien und ein neues Formatier-Programm, das mit einer Formatierzeit von 8 Sekunden ohne Verify und nur 15 Sekunden mit Verify selbst die schnellsten Floppy-Speeder überbietet.

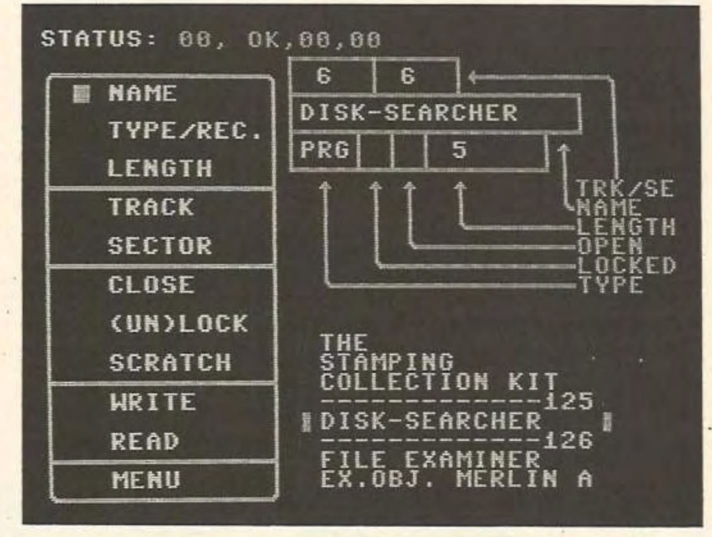


Bild 7. Der Disc-Wizard »zaubert« mit Ihren Disketten

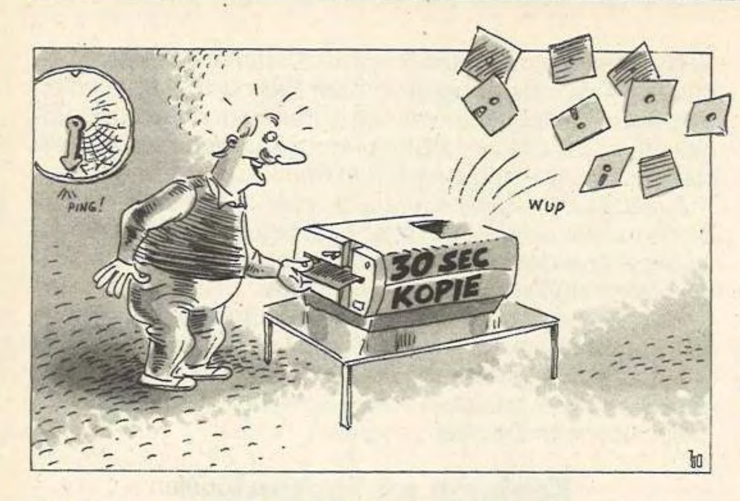


Bild 8. Beim Kopieren auf Zack. »Thirty Seconds« kopiert komplette Disketten in nur 30 Sekunden. Das Kopierprogramm wurde für die Floppy 1571 entwickelt und arbeitet mit einem Laufwerk.

Speziell für die Besitzer einer Datasette hält das Sonderheft 15 ein interessantes Kopierprogramm bereit, das Programme und andere Dateien von Diskette auf Kassette überträgt, und zwei Datasette-Beschleuniger, die mit großer Geschwindigkeit Programme laden.

Fortsetzung folgt...

Und damit befinden wir uns wieder in der Gegenwart bei Sonderheft 24, das Sie soeben lesen. Gefüllt mit interessanten Grundlagen und nützlichen Programmen knüpft es an das Ziel seiner Vorgänger an, nämlich mit umfassenden Informationen den 64'er-Lesern das Speichermedium »Floppystation« näherzubringen und die letzten Geheimnisse zu entlocken. Doch wird dieses neuste Sonderheft zur Floppy 1541 und 1571 nicht das letzte sein, da insbesondere die Floppy 1571 noch viele ungelöste Rätsel in sich birgt, die es zu entdecken gilt. (Michael Thomas/kn)

Checksummer und MSE für C 64 und C 128

Diese drei Programme sind unentbehrlich beim Abtippen unserer Listings. Sie helfen, Tippfehler vor allem bei Maschinenprogrammen zu vermeiden und sparen eine Menge Zeit.

Nobody is perfect. Jeder Computer-Fan, egal ob blutiger Anfänger oder ausgefuchster Profi, macht beim Abtippen von Programmen Tippfehler. Diese Fehler später zu finden, kann ein langwieriges Unterfangen sein.

Deshalb haben wir für Sie die Programme »Checksummer V3«, »Checksummer 128« und »MSE« (Maschinen-SpracheEditor) entwickelt. Der Checksummer ist für Basic-Programme und der MSE für Maschinensprache-Listings zuständig.

Der Checksummer

Zuerst einmal müssen Sie das Checksummer-Programm (siehe Listing 1) abtippen. Dabei sollten Sie äußerst sorgfältig vorgehen, vor allem bei den Zahlen in den DATA-Zeilen 20 bis 30. Wenn Sie trotzdem noch einen Tippfehler gemacht haben, meldet sich das Programm später mit einem entsprechenden Hinweis. Wenn Sie fertig sind, speichern Sie das Programm auf Diskette oder Kassette.

Jetzt geht es los:

1. Starten Sie den Checksummer durch die Eingabe von »RUN« und das Drücken der RETURN-Taste.
2. Wenn die Meldung »Checksummer aktiviert...« auf dem Bildschirm erscheint, haben Sie keinen Tippfehler gemacht und der Checksummer ist nun eingeschaltet.
3. Zum Löschen des Basic-Programms geben Sie bitte »NEW« ein. Keine Angst, der Checksummer selbst wird dadurch nicht gelöscht.
4. Nun können wir den Checksummer testen. Geben Sie bitte folgende Zeile ein und drücken Sie die RETURN-Taste:

```
1REM
```

In der linken oberen Bildschirmecke sehen Sie nun die Prüfsumme über die eben eingegebene Basic-Zeile. Sie muß <63> lauten. Dem Checksummer ist es übrigens egal, ob Sie »1 REM« oder »1REM« eintippen. Nur innerhalb von Anführungszeichen ist die richtige Anzahl von Leerzeichen wichtig. Diese Prüfsummen erscheinen (sofern Sie den Checksummer eingeschaltet haben) immer dann, wenn Sie eine Basic-Zeile eintippen und dann die RETURN-Taste drücken. Im 64'er-Magazin finden Sie die Prüfsumme immer am Ende jeder Programmzeile.

CTRL steht für Control-Taste, so bedeutet {CTRL+A}, daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

{DOWN}	Taste neben rechtem Shift,	{SPACE}	Leertaste	{RVSON}	Control-Taste & 9
	Cursor unten	{SHIFT-Space}	Shift-Taste & Leertaste	{RVOFF}	Control-Taste & 0
{UP}	Shift-Taste & Taste neben	{F1} bis {F8}	Funktionstasten	{ORANGE}	Commodore-Taste & 1
	rechtem Shift; Cursor hoch	{RETURN}	Return-Taste	{BROWN}	Commodore-Taste & 2
{CLR}	Shift-Taste & 2. Taste	{BLACK}	Control-Taste & 1	{LIG.RED}	Commodore-Taste & 3
	ganz rechts oben	{WHITE}	Control-Taste & 2	{GREY 1}	Commodore-Taste & 4
{INST}	Shift-Taste & Taste	{RED}	Control-Taste & 3	{GREY 2}	Commodore-Taste & 5
	ganz rechts oben	{CYAN}	Control-Taste & 4	{LIG.GREEN}	Commodore-Taste & 6
{HOME}	2. Taste von ganz rechts oben	{PURPLE}	Control-Taste & 5	{LIG.BLUE}	Commodore-Taste & 7
{DEL}	Taste ganz rechts oben	{GREEN}	Control-Taste & 6	{GREY 3}	Commodore-Taste & 8
{RIGHT}	Taste ganz rechts unten	{BLUE}	Control-Taste & 7		
{LEFT}	Shift-Taste & Taste unten rechts	{YELLOW}	Control-Taste & 8		

Tabelle 1. Die Steuerbefehle in den Listings

Diese Zahlen dürfen Sie NICHT mit abtippen.

Als Beispiel sehen Sie Bild 1. Am rechten Rand jeder Spalte sehen Sie die Prüfsummen in eckigen Klammern.

Damit sind wir beim zweiten wichtigen Punkt: Sehen Sie sich die Zeile 240 von Listing 2 genauer an. Nach dem ersten Anführungszeichen nach dem PRINT-Befehl sehen Sie eine geschweifte Klammer {}. Immer, wenn Sie in einem unserer Listings diese Klammern sehen, dürfen Sie das, was innerhalb der Klammern steht, nicht eintippen. Sie müssen die entsprechende Taste drücken. Beispiel:

```
10 PRINT "{CLR}"
```

bedeutet: Nach dem Anführungszeichen die »Bildschirmlöschen«-Taste drücken (<SHIFT CLR/HOME>). In Tabelle 1 sehen Sie eine Zusammenfassung aller möglichen Steuertasten mit dem entsprechenden Klartext.

```
10 PRINT"CHECKSUMMER FUER C 64"
11 PRINT:PRINT"EINEN MOMENT, BITTE ..."
12 FOR I=828 TO 864:READ A:POKE I,A:PS=PS+
   A:NEXT I
13 IF PS<>5765 THEN PRINT"TIPPFEHLER IN DE
   N ZEILEN 20 BIS 22":END
14 SYS 828:PS=0:FOR I=58464 TO 58583:READ
   A:POKE I,A:PS=PS+A:NEXT I
15 IF PS<>16147 THEN PRINT"TIPPFEHLER IN D
   EN ZEILEN 22 BIS 30":END
16 POKE 1,53:POKE 42289,96:POKE 42290,228
17 PRINT"CHECKSUMMER AKTIVIERT."
18 PRINT:PRINT" AUSSCHALTEN : POKE1,55 ODE
   R"SPC(27)"<RUN/STOP+RESTORE>"
19 PRINT:PRINT" ANSCHALTEN : POKE1,53"
20 DATA 169,0,133,254,162,1,189,93,3,133,2
   55,160,0,177,254
21 DATA 145,254,136,208,249,230,255,165,25
   5,221,95,3,208,238,202
22 DATA 16,230,96,160,224,192,0,160,2,169,
   0,170,133,254,177
23 DATA 95,240,40,201,32,208,3,200,208,245
   ,133,255,138,41,7
24 DATA 170,240,14,72,165,255,24,42,105,0,
   202,208,249,133,255
25 DATA 104,170,232,165,255,24,101,254,133
   ,254,76,111,228,192,4
26 DATA 48,219,198,214,165,214,72,162,3,16
   9,32,157,1,4,189
27 DATA 212,228,32,210,255,208,12,0,92,72,
   32,201,255,170,104
28 DATA 144,1,138,96,202,16,228,166,254,16
   9,0,32,205,189,169
29 DATA 62,32,210,255,104,133,214,32,108,2
   29,169,141,32,210,255
30 DATA 76,128,164,9,60,18,19
```

© 64'er

Listing 1. Der »Checksummer 64 V3« für Basic-Listings

```
5 PRINT CHR$(14)                              <242>
10 PRINT"{CLR}"                               <254>
20 PRINT"Q******************************W"    <130>
30 PRINT"{4DOWN,2SPACE}TEST{SPACE,BLUE,6SP
   ACE}"                                      <022>
40 PRINT"BBBBBBBBBBBBBBBBBBBBBBBBBBBBB"       <108>
```

© 64'er

Bild 1. Die Bedeutung der Steuerzeichen wird im nachfolgenden Text erklärt

In Zeile 10 müssen Sie nach den Anführungszeichen die Tasten <SHIFT CLR/HOME> drücken und nicht die Klammern mit dem Wort CLR eingeben. In Zeile 20 drücken Sie nach den Anführungszeichen die CBM-Taste und den Buchstaben <Q>, gefolgt von mehreren SHIFT- und Stern-Tasten und zum Schluß die CBM-Taste und den Buchstaben <W>. In Zeile 30 ist es viermal die CURSOR-abwärts-Taste, gefolgt von zweimaliger Leertaste, dann <SHIFT T> und normal EST, zum Schluß noch einmal die Leertaste, die Farbtaste Blau <CTRL 7> und sechsmal die Leertaste. Zeile 40 besteht lediglich aus mehreren Grafikzeichen, die mit der CBM-Taste und <B> erzeugt werden.

Weiterhin sehen Sie in Bild 1 (Bedeutung der Steuerzeichen) in Zeile 30 ein unterstrichenes »T« nach der Klammer. Das bedeutet, daß Sie ein »T« zusammen mit der SHIFT-Taste drücken müssen, also <SHIFT T>. Wenn ein Zeichen »überstrichen« ist, müssen Sie dieses zusammen mit der CBM-Taste eingeben. Die CBM-Taste befindet sich ganz links unten auf der Tastatur und hat die Aufschrift »C=«.

Wenn das Basic-Programm (Listing 1) einmal bis zum Ende durchgelaufen ist, können Sie den Checksummer 64 durch Eingabe von »POKE 1,55« aus-, und durch »POKE 1,53« jederzeit wieder einschalten (gilt nur für den C64). Listing 1 benötien Sie erst dann wieder, wenn Sie Ihren Computer ausgeschaltet hatten.

Es ist übrigens ratsam, vor dem Austesten eines abgetippten Programms den Checksummer 64 durch »POKE 1,55« abzuschalten. Einige Programme könnten sonst abstürzen, obwohl sie fehlerfrei abgetippt sind.

Der Checksummer 128

Zuerst einmal müssen Sie das Checksummer-Programm (siehe Listing 3) mit dem MSE im C 64-Modus abtippen. Die Eingabehinweise entnehmen Sie bitte untenstehenden Angaben zur Eingabe von MSE-Listings. Wenn Sie fertig sind, speichern Sie das Programm auf Diskette oder Kassette.

Jetzt geht es los:

1. Starten Sie nach dem Laden den Checksummer (im C 128-Modus!) durch die Eingabe von »RUN« und das Drücken der RETURN-Taste.
2. Wenn die Meldung »Checksummer 128 aktiv« auf dem Bildschirm erscheint, haben Sie keinen Tippfehler gemacht und der Checksummer ist nun eingeschaltet.

Die Bedienung des Checksummer 128 entspricht der des Checksummer 64. Einziger Unterschied: Die Prüfsummen enthalten neben den Zahlen auch Buchstaben.

Der MSE

Der MSE dient zur Eingabe von Maschinensprache-Programmen. Als erstes müssen Sie den sogenannten »MSE-Lader« (Listing 2) abtippen. Dieser erzeugt erst das eigentliche MSE-Programm auf Diskette oder Kassette.

Wichtig: Vor dem Eintippen des MSE-Laders müssen Sie unbedingt folgende Befehle eingeben (ohne Basic-Zeilennummer): POKE 44,32 : POKE 8192,0 : NEW

Jetzt können Sie beginnen, das Listing 2 abzutippen. Der MSE-Lader erkennt zwar, wenn Sie beim Eintippen der DATA-Zeilen einen Fehler gemacht haben, aber wenn Sie ganz sicher gehen möchten, sollten Sie den Checksummer vor dem Eintippen aktivieren. Die Prüfsummen für den MSE-Lader finden Sie am Ende der jeweiligen Programmzeilen.

Wenn Sie das Listing 2 nicht auf einmal abtippen möchten, müssen Sie vor jedem neuen Laden des Programms unbedingt die oben genannte POKE-Zeile eingeben!

Wenn Sie alles richtig gemacht haben und das Programm fehlerfrei abgetippt wurde, speichert es sich nach dem Starten selbst auf Diskette oder Kassette unter dem Namen »MSE V1.1«. Dieses fertige MSE-Programm laden Sie dann bei Bedarf wie ein normales Basic-Programm und starten es mit »RUN«.

So arbeitet man mit dem MSE

Als erstes möchte der MSE den Namen des zu bearbeitenden Programms wissen. Dieser steht in der ersten Zeile unserer MSE-Listings. Dann müssen Sie die Start- und Endadresse des Programms eingeben. Dies sind die letzten beiden, vierstelligen Hexadezimalzahlen in der ersten Zeile unserer Listings (siehe Listing 3).

Wenn Sie ein Programm von Diskette oder Kassette laden wollen, um an einer bestimmten Stelle weiterzutippen oder noch eine Korrektur vorzunehmen, geben Sie auf die Frage nach der Startadresse ein »L« ein. Danach müssen Sie <D> oder <T> drücken, je nachdem, ob Sie von Diskette oder Kassette (»tape«) laden möchten. Wenn das Programm unter diesem Namen nicht auf der Diskette vorhanden ist oder ein sonstiger Ladefehler vorlag, meldet sich der MSE mit »I/O-ERROR«. In diesem Fall drücken Sie <RUN/STOP RESTORE> und geben einfach noch einmal »RUN« ein.

Beim Abtippen geben Sie nach und nach die abgedruckten Buchstaben und Zahlen des jeweiligen Listings ohne die Freiräume dazwischen ein. Wenn Sie in einer Zeile einen Tippfehler gemacht haben, meldet sich der MSE sofort mit einem Brummton und der Meldung »EINGABEFEHLER«. Nach einem Druck auf die RETURN-Taste können Sie mit der DEL-Taste den Fehler korrigieren. Wenn Sie das gewünschte Programm vollständig eingegeben haben, speichert es der MSE automatisch auf Diskette oder Kassette.

Bei längeren Listings ist es unwahrscheinlich, daß Sie das komplette Programm auf einmal eingeben. Sie können Ihre bisherige Tipparbeit jederzeit durch <CTRL S> auf Diskette oder Kassette speichern und Ihr Werk später fortsetzen. Sie sollten sich dann allerdings im Heft markieren, wie weit Sie beim Abtippen gekommen sind! Später geben Sie dann nach dem Laden des ersten Programmteils <CTRL N> ein und auf die dann folgende Frage nach der Startadresse die Zeilennummer (Adresse), bei der Sie aufgehört haben zu tippen.

<CTRL M> erlaubt Ihnen jederzeit, Ihr Werk listen zu lassen. Durch <SPACE> können Sie weiterlisten lassen und durch <RUN/STOP> das Listen abbrechen.

Wenn Sie einen Drucker besitzen, können Sie das Programm auch mit <CTRL P> ausdrucken. Mit <CTRL L> wird das Programm noch einmal neu in Ihren C 64 geladen.

(F. Lonczewski/N. Mann/D. Weineck/tr)

```
100 REM DIESES PROGRAMM ERZEUGT DEN          <210>
110 REM MSE V1.1 AUF DISKETTE.               <039>
120 REM BESITZER EINER DATASETTE             <178>
130 REM MUESSEN DIE '8' AM ENDE VON          <145>
140 REM ZEILE 343 IN EINE '1' AENDERN!       <176>
150 REM                                      <212>
230 IF PEEK(44)<>32 THEN PRINT"{CLR}SIE HA
    BEN VERGESSEN, DIE POKES EINZUGE- BEN!
    ":END                                    <050>
240 PRINT"{CLR}";:DIM H(75):FOR I=0 TO 9     <042>
250 H(48+I)=I:H(65+I)=I+10:NEXT:Z=1000       <136>
260 FOR I=2048 TO 3755 STEP 20:PRINT"{HOME
    }ICH LESE ZEILE:"Z                       <253>
261 FOR N=0 TO 19:READ A$:IF LEN(A$)<>2 TH
    EN 900                                   <062>
262 IF PEEK(63)+PEEK(64)*256<>Z THEN 800     <011>
270 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))   <199>
280 D=H(H)*16+H(L):S=S+D:POKE I+N,D          <165>
290 NEXT:READ V:IF S<>V THEN 900             <139>
300 S=0:Z=Z+1:NEXT:R=PEEK(2111):H=PEEK(210
    6)                                       <126>
301 POKE 53280,R:POKE 53281,H:POKE 646,R:P
    RINT"{CLR}DIE DATA-ZEILEN SIND FEHLERF
    REI!"                                    <080>
302 PRINT"SIE KOENNEN NUN DIE FARBEN DES M
    SE"                                      <209>
303 PRINT"EINSTELLEN.":PRINT"{2DOWN,SPACE,
    RVSON}DRUECKEN SIE <1>, <2> ODER <9>     <205>
304 PRINT"{DOWN,2SPACE}<1> - RAHMEN-/SCHRI
    FTFARBE                                  <013>
305 PRINT"{2SPACE}<2> - HINTERGRUNDFARBE     <233>
306 PRINT"{DOWN,2SPACE}<9> - FARBEN UEBERN
    EHMEN                                    <158>
307 PRINT"{2DOWN}FARBE <1>:"R:PRINT"FARBE
    <2>:"H                                   <066>
308 GET A:IF A=0 THEN 308                    <210>
309 IF A=1 THEN R=(R+1)AND 15                <098>
310 IF A=2 THEN H=(H+1)AND 15                <086>
311 IF A=9 THEN 340                          <217>
312 GOTO 301                                 <034>
340 POKE 2106,H:POKE 2111,R                  <153>
342 POKE 631,19:POKE 632,13:POKE 198,2       <135>
343 PRINT"{CLR}SAVE"CHR$(34)"MSE V1.1"CHR$
    (34)",8                                  <091>
344 POKE 43,1:POKE 44,8:POKE 45,172:POKE 4
    6,14:END                                 <140>
800 PRINT"{CLR,RVSON}SIE HABEN ZEILE"Z"{LE
    FT,SPACE}VERGESSEN:":A=PEEK(646)AND 15   <124>
810 POKE 646,PEEK(53281)AND 15:PRINT"LIST"
    Z-2"-"Z+2:POKE 646,A                     <224>
820 GOTO 920                                 <082>
900 PRINT"{CLR,RVSON}SIE HABEN EINEN TIPPF
    EHLER GEMACHT:":A=PEEK(646)AND 15        <154>
910 POKE 646,PEEK(53281)AND 15:PRINT"LIST"
    Z:POKE 646,A                             <173>
920 POKE 631,19:POKE 632,17:POKE 633,13:PO
    KE 198,3:END                             <126>
1000 DATA 00,0B,08,0A,00,9E,32,30,36,31,00
     ,00,00,A2,08,A9,36,85,A4,A9, 1247       <119>
1001 DATA 08,85,A5,A9,00,85,A6,A9,B0,85,A7
     ,A0,00,B1,A4,91,A6,C8,D0,F9, 2888       <054>
1002 DATA E6,A5,E6,A7,CA,D0,F2,A9,36,85,01
     ,4C,00,B0,20,D1,B1,A9,00,8D, 2781       <096>
1003 DATA 21,D0,A9,0F,8D,20,D0,8D,86,02,A0
     ,B3,A9,74,20,FF,B1,A0,B3,A9, 2679       <089>
1004 DATA B9,20,FF,B1,A0,00,20,CF,FF,99,01
     ,02,C8,C9,0D,D0,F5,88,F0,D2, 2912       <217>
1005 DATA C0,11,90,02,A0,10,8C,00,02,20,EA
     ,B1,A0,B3,A9,CF,20,FF,B1,20, 2327       <045>
1006 DATA 8E,B4,85,FC,85,62,20,8E,B4,85,FB
     ,85,61,20,A7,B4,D0,20,A0,B3, 2864       <199>
1007 DATA A9,E5,20,FF,B1,20,8E,B4,85,60,20
     ,8E,B4,85,5F,20,A7,B4,D0,0A, 2624       <091>
1008 DATA A5,61,C5,5F,A5,62,E5,60,90,06,20
     ,43,B3,4C,3A,B0,A9,AA,A0,00, 2379       <167>
1009 DATA EA,EA,E6,FB,D0,02,E6,FC,20,3F,B2
     ,90,EF,4C,FB,B4,A2,02,86,58, 3190       <041>
1010 DATA A9,A6,A0,9D,20,F2,B1,20,E4,FF,F0
     ,FB,C9,30,90,0C,C9,47,B0,08, 2970       <231>
1011 DATA C9,3A,90,0B,C9,41,B0,07,C9,14,D0
     ,0F,4C,0B,B1,20,D2,FF,A6,58, 2322       <121>
1012 DATA 95,F7,C6,58,D0,D2,60,AE,8D,02,F0
     ,26,C9,0C,D0,03,4C,0B,B6,C9, 2685       <057>
1013 DATA 13,D0,03,4C,8B,B5,C9,0D,D0,03,4C
     ,BA,B4,C9,10,D0,03,4C,68,B5, 2282       <225>
1014 DATA C9,0E,D0,06,20,5F,B4,4C,64,B1,4C
     ,92,B0,A5,F9,20,02,B1,0A,0A, 2132       <208>
1015 DATA 0A,0A,85,F9,A5,F8,20,02,B1,05,F9
     ,60,C9,3A,90,02,69,08,29,0F, 1950       <092>
1016 DATA 60,A6,59,E0,08,90,1F,A6,58,E0,02
     ,B0,06,20,D2,FF,4C,8E,B0,C6, 2509       <188>
1017 DATA 59,A0,14,A9,92,20,F2,B1,CA,D0,FA
     ,84,57,68,68,4C,8B,B1,A6,D3, 2891       <197>
1018 DATA E0,08,B0,03,4C,92,B0,20,D2,FF,A6
     ,58,E0,02,90,09,C6,59,20,D2, 2468       <049>
1019 DATA FF,C6,58,D0,F9,4C,8E,B0,48,4A,4A
     ,4A,4A,20,59,B1,68,29,0F,C9, 2419       <035>
1020 DATA 0A,90,02,69,06,69,30,4C,D2,FF,A2
     ,FC,9A,20,D1,B1,20,48,B2,20, 2261       <073>
1021 DATA EA,B1,20,9F,B2,A5,FC,20,4E,B1,A5
     ,FB,20,4E,B1,20,ED,B1,A9,3A, 2860       <148>
1022 DATA A0,20,20,F2,B1,A9,00,85,59,20,8E
     ,B0,20,ED,B1,A4,59,20,EF,B0, 2530       <233>
1023 DATA 91,FB,C8,84,59,C0,08,90,EC,20,10
     ,B2,A9,12,20,D2,FF,20,8E,B0, 2657       <105>
1024 DATA 20,EF,B0,C5,FF,F0,0D,20,43,B3,A9
     ,14,A0,14,20,F2,B1,4C,A2,B1, 2665       <034>
1025 DATA A9,92,20,D2,FF,20,33,B2,20,E0,B2
     ,20,3F,B2,90,9F,4C,8B,B5,A9, 2648       <123>
1026 DATA 93,20,D2,FF,A2,00,A9,03,9D,00,D8
     ,9D,00,D9,9D,00,DA,9D,00,DB, 2476       <237>
1027 DATA E8,D0,EF,60,A9,0D,2C,A9,20,4C,D2
     ,FF,20,D2,FF,98,4C,D2,FF,20, 2965       <160>
1028 DATA E4,FF,F0,FB,60,84,5D,85,5C,A0,00
     ,B1,5C,F0,06,20,D2,FF,C8,D0, 3100       <077>
1029 DATA F6,60,A5,FB,85,5A,A0,00,84,5B,B1
     ,FB,18,65,5A,85,5A,90,02,E6, 2606       <156>
1030 DATA 5B,06,5A,26,5B,C8,C0,08,90,EC,A5
     ,5A,65,5B,85,FF,60,18,A5,FB, 2467       <219>
1031 DATA 69,08,85,FB,90,02,E6,FC,60,A5,FB
```

```
        ,C5,5F,A5,FC,E5,6Ø,6Ø,AØ,B3, 31Ø6     <183>
1Ø32 DATA A9,FB,2Ø,FF,B1,AØ,Ø1,B9,ØØ,Ø2,2Ø
        ,D2,FF,CC,ØØ,Ø2,C8,9Ø,F4,A9, 2692     <Ø98>
1Ø33 DATA 14,ED,ØØ,Ø2,AA,2Ø,ED,B1,CA,DØ,FA
        ,A5,62,2Ø,4E,B1,A5,61,2Ø,4E, 2457     <Ø6Ø>
1Ø34 DATA B1,2Ø,ED,B1,A5,6Ø,2Ø,4E,B1,A5,5F
        ,2Ø,4E,B1,EA,EA,EA,EA,EA,EA, 3122     <19Ø>
1Ø35 DATA EA,EA,24,5E,1Ø,Ø1,6Ø,A9,12,2Ø,D2
        ,FF,A2,28,2Ø,ED,B1,CA,DØ,FA, 27Ø3     <Ø87>
1Ø36 DATA A9,92,4C,D2,FF,A5,D6,C9,16,BØ,Ø1
        ,6Ø,A9,AØ,85,A4,A9,78,85,A6, 2945     <2Ø4>
1Ø37 DATA A9,Ø4,85,A5,85,A7,A2,13,AØ,27,B1
        ,A4,91,A6,88,1Ø,F9,CA,FØ,19, 2671     <2Ø8>
1Ø38 DATA 18,A5,A4,69,28,85,A4,9Ø,Ø2,E6,A5
        ,18,A5,A6,69,28,85,A6,9Ø,EØ, 25Ø3     <251>
1Ø39 DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9
        ,ØF,8D,18,D4,A9,ØØ,8D,Ø5,D4, 2776     <ØØØ>
1Ø4Ø DATA A9,F7,8D,Ø6,D4,A9,11,8D,Ø4,D4,A9
        ,32,8D,Ø1,D4,A9,ØØ,8D,ØØ,D4, 2413     <126>
1Ø41 DATA AØ,8Ø,2Ø,Ø9,B3,A9,1Ø,8D,Ø4,D4,6Ø
        ,A2,FF,CA,DØ,FD,88,DØ,F8,6Ø, 2914     <24Ø>
1Ø42 DATA A9,ØF,8D,18,D4,A9,2D,8D,Ø5,D4,A9
        ,A5,8D,Ø6,D4,A9,21,8D,Ø4,D4, 2385     <119>
1Ø43 DATA A9,Ø7,8D,Ø1,D4,A9,Ø5,8D,ØØ,D4,AØ
        ,FF,2Ø,Ø9,B3,A9,2Ø,8D,Ø4,D4, 225Ø     <Ø78>
1Ø44 DATA A9,ØØ,8D,Ø1,D4,8D,ØØ,D4,6Ø,38,2Ø
        ,FØ,FF,8A,48,98,48,18,AØ,Ø6, 2179     <175>
1Ø45 DATA A2,18,2Ø,FØ,FF,AØ,B4,A9,ØA,2Ø,FF
        ,B1,2Ø,12,B3,2Ø,E4,FF,FØ,FB, 2931     <Ø93>
1Ø46 DATA A2,1D,A9,14,2Ø,D2,FF,CA,DØ,FA,68
        ,A8,68,AA,18,4C,FØ,FF,ØD,ØD, 27Ø4     <Ø88>
1Ø47 DATA ØD,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,4D,41,53
        ,43,48,49,4E,45,4E,53,5Ø,52, 1144     <216>
1Ø48 DATA 41,43,48,45,2Ø,2D,2Ø,45,44,49,54
        ,4F,52,2Ø,ØD,ØD,2Ø,2Ø,2Ø,2Ø, 1Ø23     <Ø38>
1Ø49 DATA 2Ø,2Ø,2Ø,2Ø,56,4F,4E,2Ø,4E,2E,4D
        ,41,4E,4E,2Ø,26,2Ø,44,2E,57, 1128     <2Ø6>
1Ø5Ø DATA 45,49,4E,45,43,4B,ØØ,ØD,ØD,ØD,2Ø
        ,2Ø,2Ø,5Ø,52,4F,47,52,41,4D, 11Ø2     <117>
1Ø51 DATA 4D,4E,41,4D,45,2Ø,3A,2Ø,ØØ,ØD,ØD
        ,2Ø,2Ø,2Ø,53,54,41,52,54,41, 1Ø73     <Ø95>
1Ø52 DATA 44,52,45,53,53,45,2Ø,3A,2Ø,24,ØØ
        ,ØD,ØD,2Ø,2Ø,2Ø,45,4E,44,41, 1Ø14     <129>
1Ø53 DATA 44,52,45,53,53,45,2Ø,2Ø,2Ø,3A,2Ø
        ,24,ØØ,92,Ø1,Ø1,5Ø,52,4F,47, 1136     <228>
1Ø54 DATA 52,41,4D,4D,2Ø,3A,2Ø,ØØ,12,2Ø,2Ø
        ,2A,2A,2A,2Ø,46,41,4C,53,43, 1Ø24     <Ø27>
1Ø55 DATA 48,45,2Ø,45,49,4E,47,41,42,45,2Ø
        ,2A,2A,2A,2Ø,2Ø,92,ØØ,ØD,ØD, 1Ø58     <Ø98>
1Ø56 DATA 2A,2A,2A,2Ø,45,4E,44,45,2Ø,2A,2A
        ,2A,ØØ,13,Ø1,2Ø,2Ø,12,44,92, 916      <153>
1Ø57 DATA 49,53,4B,2Ø,4F,44,45,52,2Ø,12,54
        ,92,41,5Ø,45,ØD,ØØ,13,2Ø,2Ø, 1151     <Ø35>
1Ø58 DATA 49,2F,4F,2Ø,2D,2Ø,46,45,48,4C,45
        ,52,ØØ,2Ø,D1,B1,2Ø,48,B2,AØ, 16Ø6     <Ø12>
1Ø59 DATA B3,A9,CF,2Ø,FF,B1,2Ø,8E,B4,85,FC
        ,2Ø,8E,B4,85,FB,C5,61,A5,FC, 32Ø7     <251>
1Ø6Ø DATA E5,62,9Ø,23,A5,FB,C5,5F,A5,FC,E5
        ,6Ø,BØ,19,2Ø,A7,B4,DØ,14,6Ø, 286Ø     <112>
1Ø61 DATA 2Ø,A7,B4,FØ,ØC,85,F9,2Ø,A7,B4,FØ
        ,Ø5,85,F8,4C,EF,BØ,68,68,2Ø, 2749     <Ø88>
1Ø62 DATA 43,B3,4C,5F,B4,2Ø,CF,FF,C9,4C,DØ
        ,Ø9,2Ø,D1,B1,2Ø,48,B2,4C,ØB, 2372     <Ø46>
1Ø63 DATA B6,C9,ØD,6Ø,A9,ØØ,85,5E,2Ø,5F,B4
        ,2Ø,EA,B1,2Ø,ØD,B5,24,5E,3Ø, 2Ø42     <12Ø>
1Ø64 DATA Ø5,2Ø,E4,FF,FØ,FB,2Ø,E1,FF,FØ,26
        ,2Ø,9F,B2,24,5E,1Ø,Ø9,2Ø,4E, 2435     <198>
1Ø65 DATA B5,2Ø,ØD,B5,2Ø,6Ø,B5,2Ø,33,B2,2Ø
        ,3F,B2,9Ø,D7,AØ,B4,A9,28,2Ø, 219Ø     <2Ø7>
1Ø66 DATA FF,B1,2Ø,E4,FF,C9,ØD,DØ,F9,A9,ØØ
        ,85,5E,A5,61,85,FB,A5,62,85, 3Ø56     <24Ø>
1Ø67 DATA FC,2Ø,EØ,B2,4C,64,B1,A5,FC,2Ø,4E
        ,B1,A5,FB,85,FF,2Ø,4E,B1,A9, 3ØØ3     <221>
1Ø68 DATA 2Ø,AØ,3A,2Ø,F2,B1,AØ,ØØ,2Ø,ED,B1
        ,B1,FB,2Ø,4E,B1,C8,CØ,Ø8,9Ø, 2566     <Ø7Ø>
1Ø69 DATA F3,2Ø,ED,B1,24,5E,3Ø,Ø3,A9,12,2C
        ,A9,2Ø,2Ø,D2,FF,2Ø,1Ø,B2,A5, 219Ø     <Ø59>
1Ø7Ø DATA FF,2Ø,4E,B1,A9,92,2Ø,D2,FF,4C,EA
        ,B1,A9,FF,85,B8,85,B9,A9,Ø4, 3Ø73     <Ø29>
1Ø71 DATA 85,BA,2Ø,CØ,FF,A2,FF,4C,C9,FF,2Ø
        ,CC,FF,A9,FF,4C,C3,FF,2Ø,5F, 3315     <189>
1Ø72 DATA B4,A9,8Ø,85,5E,2Ø,4E,B5,2Ø,48,B2
        ,A2,24,A9,2D,2Ø,D2,FF,CA,DØ, 2596     <111>
1Ø73 DATA FA,2Ø,EA,B1,2Ø,EA,B1,2Ø,6Ø,B5,4C
        ,C1,B4,2Ø,B8,B5,A6,5F,A4,6Ø, 2812     <Ø15>
1Ø74 DATA A9,61,2Ø,D8,FF,BØ,ØA,2Ø,B7,FF,29
        ,BF,DØ,Ø3,4C,FB,B4,A9,Ø1,2Ø, 2577     <2Ø1>
1Ø75 DATA C3,FF,2Ø,68,B6,AØ,B4,A9,4F,2Ø,FF
        ,B1,2Ø,F9,B1,4C,FB,B4,2Ø,68, 2921     <237>
1Ø76 DATA B6,A9,37,AØ,B4,2Ø,FF,B1,2Ø,F9,B1
        ,A2,Ø8,C9,44,FØ,Ø6,A2,Ø1,C9, 2717     <213>
1Ø77 DATA 54,DØ,F1,A9,Ø1,A8,2Ø,BA,FF,AØ,ØØ
        ,EØ,Ø1,FØ,1A,A9,4Ø,8D,2Ø,Ø2, 24Ø3     <1Ø1>
1Ø78 DATA A9,3A,8D,21,Ø2,B9,Ø1,Ø2,99,22,Ø2
        ,C8,CC,ØØ,Ø2,9Ø,F4,C8,C8,DØ, 2182     <127>
1Ø79 DATA ØC,B9,Ø1,Ø2,99,2Ø,Ø2,C8,CC,ØØ,Ø2
        ,DØ,F4,98,A2,2Ø,AØ,Ø2,4C,BD, 2Ø18     <Ø25>
1Ø8Ø DATA FF,2Ø,B8,B5,A5,BA,C9,Ø8,9Ø,33,A6
        ,B9,86,57,A9,Ø1,2Ø,C3,FF,A9, 28ØØ     <Ø22>
1Ø81 DATA 6Ø,85,B9,2Ø,CØ,FF,BØ,28,A5,BA,2Ø
        ,B4,FF,A5,B9,2Ø,96,FF,2Ø,A5, 2911     <Ø53>
1Ø82 DATA FF,85,61,A5,9Ø,4A,4A,BØ,13,2Ø,A5
        ,FF,85,62,2Ø,AB,FF,A5,57,85, 2663     <214>
1Ø83 DATA B9,A9,ØØ,2Ø,D5,FF,9Ø,Ø3,4C,A3,B5
        ,86,5F,84,6Ø,A5,BA,C9,Ø1,DØ, 2639     <131>
1Ø84 DATA ØA,AD,3D,Ø3,85,61,AD,3E,Ø3,85,62
        ,4C,FB,B4,A9,13,2Ø,D2,FF,A2, 23ØØ     <12Ø>
1Ø85 DATA 1C,2Ø,ED,B1,CA,DØ,FA,6Ø,ØØ,ØØ,ØØ
        ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ,ØØ, 123Ø     <143>
```


Listing 2. Der MSE-Lader

```
Name : checksummer 128    1c01 1e9b
-------------------------------------
1c01 : 29 1c c3 07 fe 02 30 3a  3f
1c09 : 99 22 93 43 48 45 43 4b  53
1c11 : 53 55 4d 4d 45 52 20 31  d6
1c19 : 32 38 20 41 4b 54 49 56  c1
1c21 : 22 3a 9e 37 32 31 31 00  60
1c29 : 00 00 a9 80 85 9d a9 00  90
1c31 : 85 7f a9 ff 85 3c a9 59  74
1c39 : 85 4e a9 1c 85 4f a9 22  91
1c41 : 85 50 a9 0b 85 51 a0 00  1f
1c49 : a2 02 b1 4e 91 50 c8 d0  83
1c51 : f9 e6 4f e6 51 ca 10 f2  00
1c59 : 4c 2a 0b 00 00 00 00 00  7d
1c61 : a9 00 8d 00 ff a9 0d 20  2f
1c69 : d2 ff ad cf 41 c9 56 d0  fd
1c71 : 11 ad d0 41 c9 37 d0 0a  63
1c79 : ad d1 41 c9 2e d0 03 a9  61
1c81 : ff 2c a9 00 8d 25 0b aa  84
1c89 : d0 0b 8d 64 0d a9 65 85  8d
1c91 : 2b a9 0d 85 2c 20 cd 0b  96
1c99 : a9 6c 8d 02 03 a9 0b 8d  e1
1ca1 : 03 03 2c 27 0b 10 06 ee  3d
1ca9 : 27 0b 4c 35 0d 2c 26 0b  f0
1cb1 : 10 03 4c f9 0c 20 a4 0b  00
1cb9 : 20 af 0b 20 c2 0b aa f0  89
1cc1 : e1 b0 0c ce 26 0b 20 e3  da
1cc9 : 0b 20 af 0b 20 c2 0b 2c  ce
1cd1 : 25 0b 30 03 4c 94 a4 4c  7d
1cd9 : d7 4d 2c 25 0b 30 03 4c  dd
1ce1 : 60 a5 4c 93 4f a9 ff a2  21
1ce9 : 01 2c 25 0b 30 05 85 7a  e1
1cf1 : 86 7b 60 85 3d 86 3e 60  bf
1cf9 : 2c 25 0b 30 03 4c 73 00  e1
1d01 : 4c 80 03 2c 25 0b 30 03  45
1d09 : 4c 44 a6 4c d9 51 2c 25  ce
1d11 : 0b 30 03 4c 13 a6 4c 64  df
1d19 : 50 2c 25 0b 30 06 20 6b  b5
1d21 : a9 4c f1 0b 20 a0 50 a2  5c
1d29 : 00 2c 25 0b 10 02 a2 02  8a
1d31 : b5 15 85 aa b5 14 85 a9  8d
1d39 : 60 a9 00 85 a7 85 a8 a9  bb
1d41 : ff 85 fc a9 07 85 fd 20  4c
1d49 : 9d 0c f0 77 c9 20 f0 06  85
1d51 : 20 8e 0c 4c 11 0c 20 9d  72
1d59 : 0c f0 68 c9 20 f0 f7 c9  2e
1d61 : 3a d0 06 20 8e 0c 4c 20  44
1d69 : 0c c9 52 d0 29 20 9d 0c  2b
1d71 : c9 45 d0 1a 20 9d 0c c9  07
1d79 : 4d d0 0b 20 9d 0c f0 43  7a
1d81 : c9 3a d0 f7 f0 dd a5 fc  29
1d89 : d0 02 c6 fd c6 fc a5 fc  b0
1d91 : d0 02 c6 fd c6 fc a5 fc  b8
1d99 : d0 02 c6 fd c6 fc 20 9d  ec
1da1 : 0c f0 20 c9 20 f0 f7 c9  64
1da9 : 3a f0 b8 c9 22 f0 06 20  c5
1db1 : 8e 0c 4c 68 0c 20 8e 0c  7a
1db9 : 20 9d 0c f0 06 c9 22 d0  a2
1dc1 : f4 f0 ec 60 06 a7 26 a8  fc
1dc9 : 08 46 a7 28 26 a7 45 a7  e7
1dd1 : 85 a7 60 e6 fc d0 02 e6  4b
1dd9 : fd a0 00 b1 fc 60 ea ea  b1
1de1 : ea ea ea ea ea ea ea ea  e0
1de9 : ea a5 a8 45 a7 85 a8 20  02
1df1 : bf 0c 20 bf 0c a9 00 a2  0a
1df9 : 04 06 a7 26 a8 2a ca 10  d6
1e01 : f8 c9 0a 90 02 69 06 69  c9
1e09 : 30 4c d2 ff a9 00 8d 00  e5
1e11 : ff 38 20 f0 ff 8a 48 a9  1b
1e19 : 13 20 d2 ff a9 3c 20 d2  93
1e21 : ff 20 b3 0c a9 3e 20 d2  51
1e29 : ff 68 aa 18 4c f0 ff ad  b2
1e31 : 27 03 48 ad 26 03 48 a9  91
1e39 : 49 8d 26 03 a9 0d 8d 27  ba
1e41 : 03 2c 25 0b 10 03 a2 02  ad
1e49 : 2c a2 00 a5 a9 95 14 a5  5e
1e51 : aa 95 15 20 d8 0b a9 00  9c
1e59 : 8d 28 0b 2c 25 0b 30 08  be
1e61 : ce 27 0b a0 02 4c d8 a6  cd
1e69 : 20 f8 50 20 03 0c ee 26  b6
1e71 : 0b 68 8d 26 03 68 8d 27  d0
1e79 : 03 20 d6 0c 4c 6c 0b 85  23
1e81 : ab 8a 48 ae 28 0b a5 ab  22
1e89 : c9 0d d0 02 a9 00 9d 00  5e
1e91 : 08 ee 28 0b 68 aa a5 ab  46
1e99 : 18 60 28 4a 2f 4e 29 20  7f
```

Listing 3. Der »Checksummer 128« für die Eingabe von Basic-Listings

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Geschäftsführender Chefredakteur: Michael Scharfenberger

Chefredakteur: Albert Absmeier
Stellv. Chefredakteur: Georg Klinge
Leitender Redakteur: Gottfried Knechtel (kn)
Chef vom Dienst: Bärbel Gebhardt
Redaktion: Klaus Schrödl (sk), Ralf Sablowski (rs), Alfred Poschmann (ap)

Hotline: Gerd Donaubauer, Monika Welzel (640)

Mitarbeiter der Redaktion: Karsten Schramm, Andreas Lietz, Michael Thomas, Dr. Rudolf Egg, Florian Müller, Thomas Lipp

Redaktionsassistenz: Andrea Kaltenhauser (202)

Layout: Leo Eder (Leitung), Rolf Raß (Cheflayouter)
Andrea Miller, Katja Milles

Fotografie: Jens Jancke

Titelfoto: The Image Bank

Titelgestaltung: Erich Schulze

Produktionsleiter: Klaus Buck

Anzeigenverkaufsleitung: Ralph-Peter Rauchfuss

Anzeigenverkauf: Britta Fiebig (282)

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-41 5656, Telex: 862329
USA: M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063 Telefon: (415) 366-3600

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen und dazu, daß Markt&Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Marketingleiter: Hans Hörl (114)

Vertriebsleiter: Helmut Grünfeldt (189)

Anzeigenverwaltung und Disposition: Lisa Landthaler (233)

Druck: SOV Graphische Betriebe, Laubanger 23, 8600 Bamberg

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Preis: Das Einzelheft kostet DM 14,–

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

Verantwortlich:
Für redaktionellen Teil: Albert Absmeier
Für Anzeigen: Britta Fiebig

Redaktionsdirektor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft,
Hans-Pinsel-Straße 2, 8013 Haar bei München,
Telefon (089) 4613-0, Telex 5-22052

ISSN 0931-8933

